**Table des matières**

**Introduction Générale**

1. **Introduction à la cryptographie**
   1. Qu'est-ce que la cryptographie
   2. Objectifs Principaux de la Cryptographie
   3. L'histoire de la cryptographie
      1. **L’antiquité et les balbutiements de la cryptographie**
      2. **Moyen-Âge et Renaissance**
      3. **À l'aube de la Grande Guerre**
      4. **De la Première Guerre mondiale à l’avènement des machines de cryptage mécanique**
      5. **Méthodes de chiffrement actuelles – Le cryptage à l’ère de l’informatique et d’Interne**
2. Concepts de base de la cryptographie
   1. Les Clés en Cryptographie
   2. Chiffrement classique
      1. **Chiffrement par substitutions** 
         1. **Substitution mono alphabétique**
         2. **Substitution poly-alphabétique**
         3. **Substitution polygrammique**
      2. **Chiffrement par transposition**
   3. Chiffrement moderne
      1. **Les modes de chiffrement** 
         1. **Le mode ECB (Electronic Code Book)**
         2. **Le mode CBC (Cipher Block Chaining)**
         3. **Le mode CFB, Cipher FeedBack**
         4. **Le mode OFB, Output FeedBack**
         5. **Le mode CTR, Counter-mode encryption**
      2. **Chiffrement symétrique :**
         1. **Algorithmes symétriques populaires** 
            1. **DES**

**Caractéristiques techniques de DES**

**Avantages et limites**

* + - * 1. **Triple DES (3DES)**
        2. **AES**

**Quelques aspects techniques d'AES**

**Structure et fonctionnement d’AES**

**Sécurité et robustesse d'AES**

* + 1. **Chiffrement Asymétrique** 
       1. **Diffie-Hellman**
       2. **RSA**
          1. **Génération des clés**
          2. **Chiffrement d’un message**
          3. **Déchiffrement du message**
       3. **ElGamal**
          1. **Génération des clés**
          2. **Chiffrement**
          3. **Déchiffrement**
    2. **Comparaison entre le chiffrement symétrique et le chiffrement asymétrique**
    3. **La cryptographie hybride** 
       1. **Principe de la Cryptographie Hybride**
          1. **Efficacité et rapidité**
          2. **Sécurité dans l'échange de clés**
          3. **Combinaison des deux approches**
       2. **Exemples de Crypto-systèmes Hybrides**
          1. **PGP (Pretty Good Privacy)**

**Étapes du chiffrement avec PGP**

* + 1. **Fonction de hachage** 
       1. **Qu’est-ce qu’une fonction de hachage**
       2. **Propriétés d'une Fonction de Hachage**
       3. **L’utilisation des fonctions de hachage cryptographique**
       4. **Exemple de fonction de hachage** 
          1. **Algorithme SHA (Secure Hash Algorithm)**

**SHA-1**

**SHA-224**

**SHA-256**

**SHA-384**

* + - 1. **SHA-512**

1. Introduction au BIG DATA
   1. Définition du Big Data
   2. Type des données Big Data
      1. **Données structurées**
      2. **Données non-structurées**
      3. **Données semi-structurées**
   3. Caractéristiques de Big Data (5V du Big data)
      1. Volume
      2. **Vitesse**
      3. **Variété**
      4. **Véracité**
      5. **Valeur**
   4. **Technologies et plateformes pour Big Data** 
      1. **Hadoop**
      2. **Apache Spark**
      3. **MongoDB**
   5. Application du Big data
      1. **La santé**
      2. **Le secteur bancaire**
      3. **L'internet des objets**
      4. **Personnalisation de l'expérience client**
2. Les risques de la Sécurité dans le Big Data
   1. **Attaques passives**
   2. **Attaques actives**
   3. Quelques solutions de sécurité pour le Big Data
      1. **Chiffrement NoSQL**
   4. Défis de la Cryptographie dans le Big Data
3. Techniques de cryptographie adaptés au Big data
   1. Chiffrement homomorphe
      1. **Types de chiffrement homomorphe**
         1. **Chiffrement homomorphe à seuil (SHE)**
         2. **Chiffrement totalement homomorphe (FHE)**
      2. **Principe de base du chiffrement homomorphe**
         1. **Chiffrement**
         2. **Calcul sur les données chiffrées**
         3. **Déchiffrement**
      3. **Cas d’usage du chiffrement homomorphe**
         1. **Analyse de Big Data**
         2. **Systèmes de vote électronique**
         3. **Sécurisation des IoT (Internet of Things)**
   2. Chiffrement par attribut
      1. **Les variantes principales du chiffrement par attributs(ABE)**
         1. **La variante KP-AB**
         2. **La variante CP-AB**
         3. **La comparaison entre ABE, KP-ABE et CP-ABE**
   3. Fonctionnement de chiffrement par attribut
      * 1. **Concept de base**
        2. **Étapes de fonctionnement**
      1. **Cas d’usage du chiffrement par attribut**
         1. **Analyse de Big Data**
         2. **Partage sécurisé des données dans des environnements multi-utilisateurs**
         3. **Protection des données sensibles dans la santé (Health Big Data)**

#### Analyse de données financières

#### Avantages dans le Big Data

#### Limites potentielles et solutions

* 1. Algorithmes de chiffrement légers
     1. **Caractéristiques des algorithmes de chiffrement léger**
     2. **Exemples d'algorithmes de chiffrement léger**

#### SPECK (Symmetric Privacy-Enhanced Communication)

* + - 1. **SIMON**
      2. **XTEA (Extended Tiny Encryption Algorithm)**
      3. **Chiffrement à clé elliptique (ECC)**
      4. **LEA (Lightweight Encryption Algorithm)**
    1. **Cas d’usage Les algorithmes de chiffrement léger**
       1. **Analyse de Big Data**

1. Distribution des bases de données
   1. Introduction à la distribution des bases
      1. **Définition des bases de données distribuées**
      2. **Types des bases de données distribuées** 
         1. **Base de données homogène**
         2. **Base de données hétérogène**
      3. **Le rôle dans le Big Data** 
         1. **Stockage des données massives**
         2. **Traitement parallèle**
         3. **Tolérance aux pannes**
         4. **Scalabilité horizontale**
         5. **Proximité des données**
      4. **Importance dans le contexte du Big Data** 
         1. **Efficacité dans la gestion des 3V du Big Data**
         2. **Support pour les technologies Big Data**
         3. **Optimisation des coûts**
         4. **Flexibilité et adaptabilité**
         5. **Amélioration de la prise de décision**
   2. Les principes fondamentaux de la distribution des bases
      1. **Architecture des bases distribuées**
      2. **Architecture de la répartition des données**
      3. **Répartition des données**
      4. **Fragmentation des données** 
         1. **Définition**
         2. **Objectif de la fragmentation**
         3. **Les problèmes de la fragmentation**
         4. **Techniques de fragmentation**

Introduction Générale

Dans un monde où les données numériques jouent un rôle central dans tous les aspects de la vie quotidienne, la sécurité des informations échangées devenue une nécessité incontournable. Les transactions financières, les communications personnelles et le stockage de données sensibles requièrent des mécanismes robustes pour garantir leur confidentialité, leur intégrité et leur authenticité.

Avec l'essor du Big Data, où des volumes gigantesques de données sont produits, échangés et analysés en temps réel, les défis de sécurité prennent une dimension sans précédent. Les systèmes traditionnels de protection ne suffisent plus face à la diversité des menaces et aux exigences de performance imposées par ces environnements massivement distribués.

La cryptographie s’impose alors comme une solution essentielle pour répondre à ces besoins. En constante évolution, elle offre des techniques capables de sécuriser des données à grande échelle, tout en restant performantes et adaptables. Ce projet explore comment les outils cryptographiques peuvent être utilisés pour garantir la sécurité des données massives dans le contexte du Big Data, tout en prenant en compte les spécificités et les enjeux liés à cette ère de l'information.

1. Introduction à la cryptographie :
   1. Qu'est-ce que la cryptographie ?

Le mot Cryptographie provient étymologiquement de : kruptos (κρυπτός) « caché » et graphein (γράφειν) « écrire ».

La cryptographie est la science et l’art de sécuriser les informations en les rendant inintelligibles à toute personne non autorisée. Elle repose sur des procédés mathématiques pour transformer un texte en clair en un texte chiffré, appelé cryptogramme, à l’aide d’une clé de chiffrement.Ce processus permet de protéger la confidentialité, l’intégrité et l’authenticité des données, que ce soit pour leur stockage ou leur transmission sur des canaux non sécurisés, tels qu’Internet.

La cryptographie implique l’utilisation d’algorithmes spécifiques pour le chiffrement et le déchiffrement, avec une dépendance cruciale à des clés secrètes de petite taille comparées aux volumes d’information qu’elles protègent. Elle constitue une branche fondamentale de la cryptologie, qui englobe également la cryptanalyse, la science visant à casser ou à analyser ces systèmes.

* 1. Objectifs Principaux de la Cryptographie :

La cryptographie permet de résoudre quatre problèmes différents :

**-La confidentialité** : Le texte chiffré ne doit être lisible que par les destinataires légitimes. Il ne doit pas pouvoir être lu par un intrus.

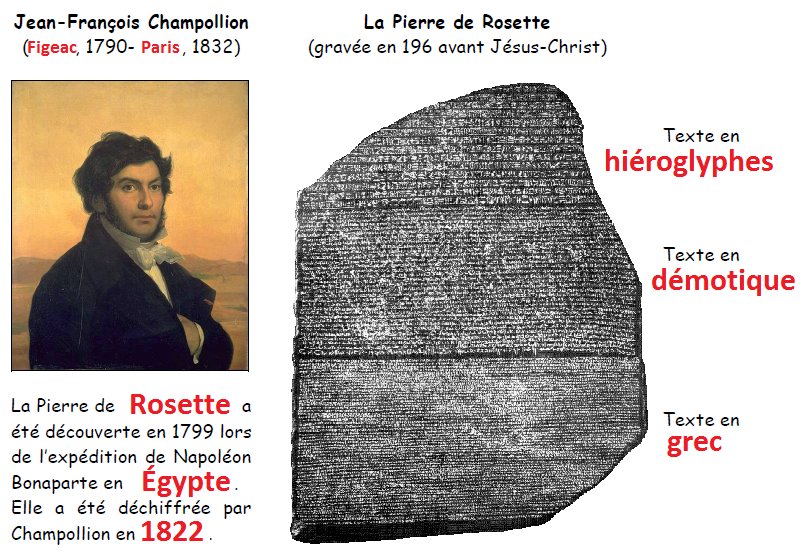
**-L'authentification :** Le destinataire d'un message doit pouvoir s'assurer de son origine. Un intrus ne doit pas être capable de se faire passer pour quelqu'un d'autre.

**-L'intégrité :** Le destinataire d'un message doit pouvoir vérifier que celui-ci n'a pas été modifié en chemin. Un intrus ne doit pas être capable de faire passer un faux message pour légitime.

**-La non répudiation :** Un expéditeur ne doit pas pouvoir, par la suite, nier à tort avoir envoyé un message.

* 1. L'histoire de la cryptographie :
     1. **L’antiquité et les balbutiements de la cryptographie :**
* **Les Hiéroglyphes :**

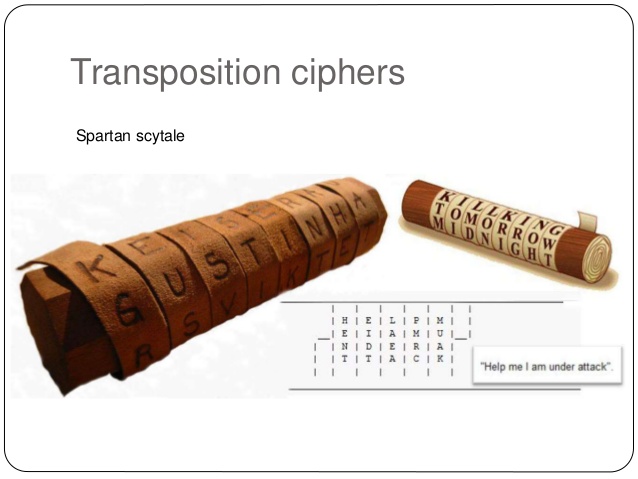
Les plus anciens chiffrements connus se présentent sous la forme de hiéroglyphes retrouvés sur des monuments datant de près de 3 000 ans avant J.C. Longtemps, les hiéroglyphes furent considérés comme indéchiffrables, avant que la découverte de la célèbre pierre de Rosette et le travail de Jean-François Champollion ne permettent d’en percer les mystères.

****

**Figure 1-** pierre de rosette.

* **La Scytale Spartiate :**

Chez les grecs, vers 400 av. J.-C., dans la cité de Sparte, l’acheminement discret de l’information se faisait grâce à un dispositif ingénieux appelé la scytale spartiate. Ce procédé consistait à enrouler une bande de cuir longue et fine autour d’un bâton de diamètre spécifique, puis à y écrire le message en texte clair, complété par des caractères aléatoires. Une fois le message rédigé, la bande était portée par le messager, généralement comme une ceinture. Pour déchiffrer le message, le destinataire devait posséder un bâton du même diamètre, permettant ainsi de dérouler correctement le cuir et de reconstituer le texte. Le diamètre du bâton constituait ainsi la clé du code.

****

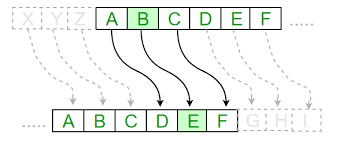
**Figure 2-**la scytale spartiate.

* **Le Chiffre de César :**

Plus tard, 60- 50 avant J.C., on assista à l’émergence du chiffre de César, d’où son nom, est l’une des méthodes de cryptage les plus célèbres de l’Histoire. Utilisé par Jules César pour sécuriser ses communications avec ses généraux, ce système repose sur un principe simple mais efficace : chaque lettre du message original est substituée par une autre située à une distance fixe dans l’alphabet. Cette distance, souvent de trois lettres, était connue à la fois de l’expéditeur et du destinataire.L’alphabet de substitution utilisé par le système de chiffrement de César est indiqué ci-dessous.

**Alphabet :**ABCDEFGHIJKLMNOPQRSTUVWXYZ

**Substitution :** DEFGHIJKLMNOPQRSTUVWXYZABC

****

* + 1. **Moyen-Âge et Renaissance :**

Le Moyen Âge marque une étape cruciale dans l’évolution des technologies cryptographiques. C’est durant cette période que les premiers systèmes de chiffrement ont été décryptés, ce qui a conduit à l’émergence de nouvelles méthodes de cryptage. En parallèle, l’augmentation des échanges diplomatiques a entraîné un volume croissant d’informations confidentielles, favorisant ainsi le recours à la cryptographie pour assurer la sécurité de ces communications.

* **La méthode de Porta :**

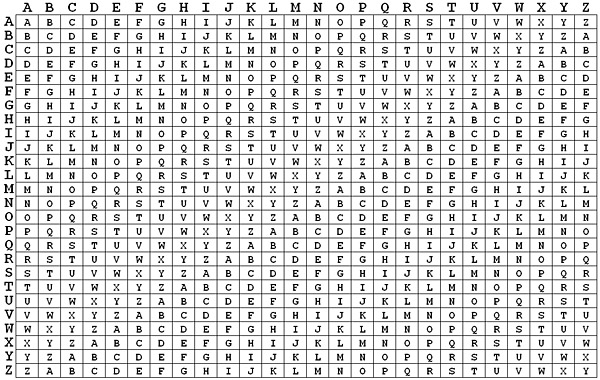
Cette méthode a été développée en 1563 par l'Italien Giovanni Battista da Porta, repose sur l'utilisation d'un mot-clé pour effectuer des substitutions de lettres. Le système est décrit au moyen de la table ci-dessous. Le mot-clé, composé de lettres-clés, est essentiel à cette méthode. La première colonne de la table (comportant des paires de lettres en gras) représente le mot-clé, tandis que la rangée du haut, également en gras, correspond au texte en clair. L'association des lettres dans la table permet une substitution réciproque pour chaque lettre-clé. Pour effectuer la substitution, si la lettre du texte en clair se trouve dans la rangée du haut, on remplace cette lettre par celle qui apparaît à l'intersection de la colonne contenant la lettre en clair et de la rangée de la lettre-clé. Si la lettre du texte en clair ne se trouve pas dans cette rangée, on la recherche dans la rangée correspondant à la lettre-clé et on lui substitue la lettre correspondante.

****

**Figure 3-**La table de Porta.

* **Le chiffre de Vigenère :**

Le chiffre de Vigenère, développé par Blaise de Vigenère en 1585, repose sur une méthode de substitution polyalphabétique. Cette technique utilise une grille, appelée la table de Vigenère, qui permet de chiffrer un message en recourant à plusieurs alphabets de substitution. Pour chiffrer un message, on commence par aligner les lettres du texte clair avec les lettres d'une clé. Chaque lettre du texte clair est remplacée par la lettre située à l'intersection de la ligne correspondant à la lettre de la clé et de la colonne correspondant à la lettre du texte clair dans la table. Cette méthode fut considérée comme indéchiffrable pendant longtemps.



**Figure 4**-table de vigenère

* + 1. **À l'aube de la Grande Guerre :**
* **Chiffre de Playfair :**

La méthode de Playfair, développée par Charles Wheatstone, professeur de philosophie au King’s College de Londres, a été nommée ainsi en 1854 lorsqu'elle fut présentée par Lyon Playfair, baron de St. Andrews, lors d'un banquet. Ce système de chiffrement utilise une grille 5x5 générée à partir d'un mot-clé et repose sur l'encodage de paires de lettres selon des règles définies par leur position dans la grille.Le chiffrement de Playfair fut utilisé par les Britanniques pendant la guerre des Boers et la Première Guerre mondiale.

* + 1. **De la Première Guerre mondiale à l’avènement des machines de cryptage mécanique :**

La Première Guerre mondiale fut un véritable catalyseur des communications cryptographiques, et donc de la cryptanalyse.

* **Le télégramme Zimmermann :**

En janvier 1917, les Britanniques démontrèrent l'importance stratégique de la cryptanalyse en interceptant et en décryptant le télégramme Zimmermann grâce au travail du « Bureau 40 », une unité spécialisée dans la cryptanalyse. Ce message chiffré, envoyé par l'Allemagne via des lignes internationales, utilisait un code que les cryptanalystes britanniques parvinrent à casser, révélant un plan allemand visant à inciter le Mexique et le Japon à attaquer les États-Unis. Ce décryptage mit en lumière le rôle crucial de la cryptographie dans les communications militaires et diplomatiques

* **Chiffre ADFGVX :**

En 1918, les Allemands introduisirent le chiffre ADFGVX, conçu par le colonel Fritz Nebel. Ce système, basé sur le carré de Polybe, utilisait les lettres A, D, F, G ,V et X comme en-têtes des lignes et colonnes. Chaque lettre du texte clair était représentée par une paire de lettres cryptées issues de cette table. Une étape supplémentaire de chiffrement par transposition était ensuite appliquée aux lettres obtenues, renforçant la sécurité du système. Cette méthode évolua rapidement en ADFGVX, une version améliorée intégrant une ligne et une colonne supplémentaires. Le choix des lettres ADFGVX s’explique par leur facilité de distinction dans les transmissions en code Morse, réduisant ainsi les risques d'erreurs.

****

**Figure 5-**La table ADFGVX

* **La machine Enigma :**

La machine Enigma, conçue par l'ingénieur allemand Arthur Scherbius en 1918, marque une avancée majeure dans la cryptographie mécanique grâce à son système de chiffrement par substitution polyalphabétique. Elle intégrait plusieurs composants essentiels, notamment des rotors codeurs munis de 26 connexions internes permettant des permutations secrètes des lettres, un brouilleur, et un tableau de connexions introduisant des substitutions supplémentaires pour accroître la complexité du chiffrement. Lorsqu'une lettre était saisie au clavier, un courant électrique traversait ces différents éléments avant d'illuminer la lettre chiffrée correspondante. Les rotors avançaient automatiquement après chaque frappe, modifiant ainsi la clé de chiffrement à chaque étape. L’Enigma utilisait un chiffrement réciproque, ce qui signifiait qu’un même paramétrage initial permettait à la fois le chiffrement et le déchiffrement des messages. La machine a évolué au fil des années, avec l'ajout de fonctionnalités comme des rotors supplémentaires et des configurations élargies, augmentant sa complexité. Les concepts et mécanismes derrière l'Enigma continuent d'influencer les systèmes modernes de cryptographie, tout en constituant une étude précieuse pour comprendre les bases et les défis des méthodes de chiffrement avancées.



**Figure 6-**machine ENIGMA

* + 1. **Méthodes de chiffrement actuelles – Le cryptage à l’ère de l’informatique et d’Internet :**

Depuis la Seconde Guerre mondiale, le cryptage et le décryptage ont évolué, passant des systèmes mécaniques aux solutions numériques. En dehors des applications militaires traditionnelles, l'essor de l'informatique dans le secteur privé a généré une demande croissante pour le cryptage des transactions commerciales et d'autres usages civils.

* **Diffie-Hellman:**

En 1976, Whitfield Diffie et Martin Hellman révolutionnent la cryptographie en introduisant le concept de cryptographie à clé publique dans leur article New Directions in Cryptography. Ils proposent une méthode innovante pour l’échange sécurisé de clés et l’utilisation de fonctions à sens unique pour l’authentification. Leur travail, fondé sur des principes mathématiques, jette les bases des protocoles modernes de sécurité.

* **DES:**

L'algorithme DES (Data Encryption Standard) fut approuvé par le NBS (National Bureau of Standards) en 1976 et devint rapidement le standard mondial de chiffrement des données. Cet algorithme, dérivé du chiffre Lucifer de Feistel dans sa version à 64 bits, a été adopté pour répondre à la nécessité de standardiser la cryptographie civile et réduire les coûts élevés liés à des systèmes personnalisés.En 1999, le DES a été déclaré obsolète en raison de vulnérabilités découvertes, ce qui a marqué la fin d'une époque et l'apparition de standards plus modernes.

* **RSA:**

L'algorithme RSA, créé en 1977 par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman au MIT, tire son nom des initiales de ses inventeurs. Cet algorithme repose sur la difficulté de la factorisation des grands nombres premiers, un principe fondamental de la cryptographie à clé publique. Bien qu'un cryptographe britannique ait développé une méthode similaire trois ans auparavant, ses travaux étaient classés secret défense et ne furent rendus publics qu'en 1997. Breveté par la société RSA Data Security, RSA est devenu un standard mondial pour la cryptographie et l'authentification, reconnu pour sa fiabilité et sa sécurité.

* **IDEA:**

En 1990, Xuejia Lai et James Massey publièrent une proposition visant à remplacer le DES par un nouvel algorithme de chiffrement, l'IDEA (International Data Encryption Algorithm). L'IDEA utilise une clé de 128 bits et repose sur des opérations adaptées à tous types d'ordinateurs, offrant ainsi une programmation plus efficace. Cet algorithme est considéré comme l'un des meilleurs en matière de sécurité, et à ce jour, personne n'a réussi à casser un bloc de texte chiffré par IDEA. L'IDEA est actuellement exploité par la société Mediacrypt.

* **La Cryptographie Quantique :**

En 1990, Charles H. Bennett et Gilles Brassard publièrent leurs résultats expérimentaux sur la cryptographie quantique, qui utilise des photons pour transmettre des bits servant à générer des clés pour des systèmes de chiffrement, tels que le chiffre de Vernam. En s'appuyant sur les lois de la mécanique quantique, cette méthode garantit non seulement la confidentialité des communications, mais permet également de détecter toute tentative d'écoute de la ligne. Toutefois, un inconvénient majeur de la cryptographie quantique réside dans sa dépendance à des câbles en fibres optiques entre les deux parties, limitant ainsi sa portée.

* **PGP :**

En 1991, le scientifique américain Philip R. Zimmermann lance la première version de PGP (Pretty Good Privacy), un logiciel de chiffrement utilisant la clé publique, destiné à protéger les communications privées. PGP combine chiffrement symétrique et par clé publique, et inclut également une fonctionnalité de signature numérique, garantissant l'intégrité des données et l'authenticité de l'expéditeur. En 1992, grâce à la collaboration d'une communauté internationale de développeurs, la version PGP 2.0 est lancée hors des États-Unis. En 1993, PGP devient le standard mondial de cryptographie, particulièrement utilisé pour sécuriser les échanges par email.

* **AES :**

En 1997, le NIST(National Institute of Standards and Technology) a lancé un appel d'offres pour la création d'un nouvel algorithme de chiffrement par blocs, recevant 50 propositions. En 2000, il a sélectionné l'algorithme Rijndael, qu'il a renommé AES (Advanced Encryption Standard). L'AES, un algorithme de chiffrement symétrique, est devenu la norme de chiffrement adoptée par le gouvernement des États-Unis. Aujourd'hui, AES est largement utilisé dans les protocoles de sécurité informatique et est reconnu comme l'un des algorithmes les plus sûrs.

1. Concepts de base de la cryptographie :
   1. Les Clés en Cryptographie :

Une clé en cryptographie est une valeur utilisée dans un algorithme pour chiffrer ou déchiffrer des données. C'est un nombre complexe dont la taille est mesurée en bits. Par exemple, une clé de 1024 bits correspond à une valeur numérique très grande. Plus la clé est longue, plus elle améliore la sécurité du système. Cependant, la sécurité optimale résulte de la combinaison d'algorithmes complexes et de clés robustes. Les clés doivent être stockées de manière sécurisée, accessible uniquement à leur propriétaire, pour garantir leur confidentialité et intégrité.

Il existe deux types principaux de clés en cryptographie :

***- Clés symétriques :*** La même clé est utilisée pour le chiffrement et le déchiffrement des données. Ce système est également appelé chiffrement à clé secrète.

*-* ***Clés asymétriques :*** Ce système utilise deux clés distinctes : une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Ce type de chiffrement est également appelé chiffrement à clé publique.

* 1. Chiffrement classique :
     1. **Chiffrement par substitutions :**
        1. **Substitution mono alphabétique :**

La substitution mono alphabétique est une méthode de chiffrement où chaque lettre d'un message est remplacée de façon fixe par une autre lettre ou un symbole selon un alphabet de substitution prédéfini. Parmi ses variantes figurent le chiffre de César, les alphabets désordonnés et le chiffre affine. Simple à appliquer, cette technique remplace chaque lettre par une autre de manière constante, rendant le message chiffré lisible uniquement avec la clé correspondante. Cependant, elle est vulnérable à l'analyse fréquentielle, car les lettres les plus fréquentes d'une langue restent repérables dans le texte chiffré, ce qui en limite la sécurité face aux outils modernes.

* + - 1. **Substitution poly-alphabétique :**

Les substitutions poly-alphabétiques, aussi appelées à double clé ou à alphabets multiples, utilisent plusieurs alphabets pour chiffrer un message, ce qui permet à une même lettre d'être remplacée par différents symboles. Un exemple célèbre est le chiffre de Vigenère, qui a résisté aux cryptanalystes pendant trois siècles. Cette méthode devient plus résistante en utilisant une suite de chiffres monoalphabétiques réutilisée périodiquement, rendant les attaques plus difficiles par rapport à une simple substitution avec un seul alphabet.

* + - 1. **Substitution polygrammique:**

Les substitutions polygrammiques, aussi appelées polygraphiques, sont une méthode de chiffrement où des groupes de lettres (généralement deux ou trois) sont remplacés par des symboles correspondants. Contrairement à la substitution mono alphabétique où chaque lettre est chiffrée individuellement, cette méthode chiffre des digrammes (groupes de deux lettres) ou trigrammes. Par exemple, dans un système comme celui de Playfair, des paires de lettres sont transformées en paires de symboles. De même, dans le système de Hill, une transformation mathématique est utilisée pour chiffrer ces groupes de lettres, augmentant ainsi la sécurité par rapport à la substitution simple.

* + 1. **Chiffrement par transposition :**

La transposition est un procédé de chiffrement où l'ordre des lettres d'un message est modifié sans altérer les lettres elles-mêmes, créant ainsi un mélange incohérent. Bien que cette méthode soit peu résistante aux attaques statistiques, elle devient plus complexe à déchiffrer à mesure que la longueur du message augmente. Pour des messages courts, la méthode est vulnérable, car il y a peu de variantes possibles, mais pour des messages plus longs, de nombreuses permutations sont possibles. Cependant, pour réussir à déchiffrer, il est nécessaire de connaître le procédé de transposition convenu entre les parties.

* 1. Chiffrement moderne :
     1. **Les modes de chiffrement :**
        1. **Le mode ECB (Electronic Code Book) :**

Le mode ECB est le mode le plus simple. Le message est découpé en blocs et chaque bloc est crypté séparément.

ci = Ek(mi)

* **Avantage :** Ce mode permet le chiffrement en parallèle des différents blocs composant un message.
* **Inconvénient :** Même bloc de message en clair sera toujours chiffré en un même bloc de message chiffré. De plus, un autre inconvénient qui du chiffrement par blocs est l'amplification d'erreur : si un bit du message chiffré est modifié pendant le transfert, tout le bloc de message en clair correspondant sera faux.
* La procédure ECB n’est jamais utilisée en pratique.
  + - 1. **Le mode CBC (Cipher Block Chaining) :**

Ce mode a été introduit pour qu’un bloc ne soit pas codé de la même manière s’il apparaît dans deux messages différents ou s’il apparaît deux fois dans un message. Le message, M, est découpé en blocs, (mi)i≥1, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial c. Chaque bloc clair mi est d’abord modifié en faisant un XOR de ce bloc avec le bloc crypté précédent, ci−1 puis on crypte le résultat obtenu par XORisation avec la clé.

ci = Ek(mi ⊕ ci−1)

* **Avantage :** La structure du message en clair est masquée par le chaînage. Un attaquant ne peut plus manipuler le cryptogramme, excepté en retirant des blocs au début ou à la fin.
* **Inconvénient :** On pourrait craindre que le chaînage de bloc n'entraîne une propagation d'erreurs importantes. De fait, une erreur d'un bit sur le message en clair affecte tous les blocs chiffrés suivants. Par contre, si un bit du message chiffré est modifié au cours du transfert, seul le bloc de message en clair correspondant et un bit du bloc de message en clair suivant seront endommagés : le mode CBC est dit auto réparateur.
  + - 1. **Le mode CFB, Cipher FeedBack:**

Le mode CFB, Cipher FeedBack, a été introduit pour ne pas avoir à calculer la fonction inverse, Dk, de la fonction de chiffrage Ek. Le principe est le même que celui du mode CBC. Le message, M, est découpé en blocs, (mi)i≥1, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial m0, choisi suivant les mêmes principes que le bloc c0 en mode CBC.Chaque bloc clair mi est XORé avec le crypte du bloc de sortie précédent.

ci = mi ⊕ Ek(ci−1)

* **Avantage :** Le mode CFB offre un déchiffrement rapide.
* **Inconvénient :** Il est moins sécurisé que le mode CBC et peut propager les erreurs sur plusieurs blocs si un bit du message chiffré est modifié, compromettant ainsi l'intégrité des données.
  + - 1. **Le mode OFB, Output FeedBack:**

Le mode OFB est une variante de CFB qui permet d’avoir un cryptage et un décryptage totalement symétrique :

zi = Ek(zi−1); ci = mi ⊕ zi

* + - 1. **Le mode CTR, Counter-mode encryption:**

Ce mode de cryptage est lui aussi totalement symétrique, mais en outre facilement parallélisable. Il utilise pour le chiffrement un compteur de valeur initiale T.

ci = mi ⊕ Ek(T+i)

* **Avantage :** Il offre un chiffrement parallèle, améliorant ainsi la performance, et convient bien aux données de longueur variable.
* **Inconvénient** : Il présente un risque si les compteurs sont mal gérés, car leur répétition pourrait compromettre la sécurité. De plus, toute modification d’un bit du texte chiffré affecte l’intégrité du message déchiffré.
  + 1. **Chiffrement symétrique :**

Le chiffrement symétrique transforme un texte clair en texte chiffré et permet de le déchiffrer à l’aide d’une clé unique partagée entre l’expéditeur et le destinataire. La sécurité repose sur la confidentialité de cette clé, tandis que l’algorithme reste public. Bien que rapide et efficace, cette méthode présente une faiblesse majeure qui est l’échange sécurisé de la clé entre les parties.

Les algorithmes symétriques sont de deux types :

**- Algorithmes de chiffrement en continu :** qui agissent sur le message en clair un bit à la fois. Le principe consiste à générer un flux pseudo aléatoire et de le combiner avec l’information bit à bit par l’opération XOR. A la réception, on applique le même mécanisme, et on restitue l’information.

**- Algorithmes de chiffrement par bloc :** qui opèrent sur le message en clair par groupes de bits appelés bloc. La taille typique des blocs est 64 bits, ce qui est assez grand pour interdire l’analyse et assez petit pour être pratique.

* + - 1. **Algorithmes symétriques populaires :**
         1. **DES :**

Le Data Encryption Standard (DES) est un système de chiffrement basé sur un schéma de Feistel, composé de 16 itérations d'une fonction appelée "fonction d'étage". Ce processus mélange les bits du texte clair en respectant les principes de confusion (masquer les relations entre texte clair et texte chiffré) et de diffusion (répartir la redondance du texte clair sur le texte chiffré). DES utilise des blocs de 64 bits et une clé de 56 bits (avec 8 bits supplémentaires dédiés à un code correcteur).

**Caractéristiques techniques de DES :**

**a)** **Clé et blocs** : Une clé de 56 bits est dérivée pour produire 16 sous-clés de48 bits utilisées dans chaque tour de chiffrement.

**b) Permutation initiale :** Le texte clair de 64 bits subit une permutation initiale (IP) avant d'être divisé en deux parties :32 bits de gauche (L0) et 32 bits de droite (R0).

**c) Itérations :** Sur 16 étapes, le texte est transformé à l'aide de la règle : -Li=Ri−1.-Ri=Li−1⊕ f (Ri−1, Ki), où f est une fonction non linéaire qui participe à la diffusion et qui est décrite par les S-boîtes.

**d) Permutation inverse :** Après les 16 tours, une permutation inverse (IP−1) est appliquée pour obtenir le texte chiffré final.

**Avantages et limites :**

* **Avantages :** La méthode itérée et l'usage des principes de confusion et de diffusion offrent une bonne sécurité pour son époque.
* **Limites :** Avec une clé de 56 bits, DES est vulnérable aux attaques par force brute. Dès 1997, des attaques ont permis de casser DES en quelques semaines, puis en moins d'une journée en 1999.
  + - * 1. **Triple DES (3DES) :**

Pour renforcer la sécurité de DES en attendant l'AES, le Triple DES a été introduit. Il applique successivement DES avec une clé K, le décrypte avec une clé K′, puis le rechiffre avec K. Cela équivaut à utiliser une clé plus longue, prolongeant la durée de vie de DES.

* + - * 1. **AES :**

L'Advanced Encryption Standard (AES) est un système cryptographique basé sur un réseau de substitution-permutation, contrairement au schéma de Feistel utilisé dans DES. Il repose sur une série d’opérations de permutation et de substitution, appliquées de manière répétée sur des blocs de 128 bits. AES utilise des clés de longueur variable : 128, 192 ou 256 bits, offrant une sécurité renforcée contre les attaques par force brute.

La taille de bloc fixe de 128 bits a été choisie parmi des configurations initiales qui varient entre 4 et 8 blocs de 32 bits. Grâce à l'utilisation d'une clé minimale de 128 bits, AES rend les recherches exhaustives de clés irréalisables dans un futur prévisible. Par exemple, même avec un algorithme capable de tester 256 clés par seconde (cassant DES en une seconde), il faudrait environ 149 mille milliards d'années pour casser AES.

**Quelques aspects techniques d'AES :**

AES est un algorithme de chiffrement itératif qui effectue un ensemble d’opérations sur des blocs de 128 bits à travers Ne étapes (ou tours), où Ne varie selon la longueur de la clé :

-10 tours pour une clé de 128 bits.

-12 tours pour une clé de 192 bits.

-14 tours pour une clé de 256 bits.

**Structure et fonctionnement d’AES :**

**a) Diversification des clés :** À partir de la clé secrète K, un algorithme de diversification génère Ne sous- clés de 128 bits chacune.

**b) Étapes principales :** Chaque étape applique une fonction g constituée de quatre opérations :

***1) SubBytes*** : Substitution non linéaire des octets à l’aide de S-boîtes.

***2) ShiftRows :*** Permutation circulaire des éléments des lignes de la matrice représentant les blocs.

***3) MixColumns* :** Transformation linéaire sur les colonnes de la matrice (sauf au dernier tour).

***4) AddRoundKey :***  Ajout de la sous clé correspondante (Ki) par XOR.

**c) Dernier tour :** Pour le dernier tour, l’opération MixColumns est omise.

**Sécurité et robustesse d'AES :**

Grâce à l'utilisation de blocs de 128 bits, à des clés plus longues et à un schéma complexe, AES garantit un haut niveau de sécurité. Il est conçu pour résister aux attaques les plus avancées, notamment les attaques par force brute et par cryptanalyse différentielle ou linéaire.

* + 1. **Chiffrement Asymétrique :**

Le chiffrement asymétrique, également appelé cryptographie à clé publique, est un système cryptographique qui repose sur l’utilisation de deux clés distinctes mais mathématiquement liées : une clé publique et une clé privée. La clé publique est accessible à tous et sert à chiffrer les données, tandis que la clé privée, qui doit rester secrète, est utilisée pour déchiffrer les données. Ce mécanisme permet de garantir la confidentialité, l'authenticité, et parfois la non-répudiation des messages.

* **Algorithmes asymétrique populaires :**
  + - 1. **Diffie-Hellman :**

Le protocole Diffie-Hellman permet à deux parties, A et B, d’établir une clé secrète partagée K en communiquant sur un canal non sécurisé. Il repose sur la difficulté du problème du logarithme discret, qui rend pratiquement impossible de retrouver une valeur x à partir de y=gxmod n si n est suffisamment grand. Cela garantit que même si un tiers intercepte les échanges, il ne peut pas calculer la clé partagée.

Fonctionnement :

* Choix des paramètres publics :

Un nombre premier p et un générateur g sont choisis et rendus publics.

* Choix des clés privées par les parties :

A choisit une clé privée a (un entier aléatoire), avec 1≤a<p.

B choisit une clé privée b (un entier aléatoire), avec 1≤b<p.

Ces clés privées sont gardées secrètes.

* Calcul des clés publiques :

A calcule sa clé publique :A=ga mod p.

B calcule sa clé publique :B=gb mod p.

Les clés publiques A et B sont échangées entre A et B.

-Calcul de la clé partagée :

A utilise la clé publique B pour calculer la clé partagée K : K=Ba mod p=(gb)a mod p.

B utilise la clé publique A pour calculer la clé partagée K : K=Ab mod p=(ga)b mod p.

Grâce à la propriété des exponentiations modulaire, on a :K=gab mod p.

* + - 1. **RSA:**

Le système RSA (Rivest, Shamir, Adleman) est un algorithme de cryptographie asymétrique, inventé en 1977, qui repose sur la difficulté de la factorisation des grands nombres entiers en facteurs premiers. C'est l'un des premiers systèmes robustes de chiffrement à clé publique et reste largement utilisé pour sécuriser les échanges de données sur Internet.Voici les étapes de son fonctionnement :

* + - * 1. **Génération des clés :**

Cette étape est réalisée par l'entité qui souhaite recevoir des messages chiffrés.

-Choix de deux nombres premiers distincts :

Sélectionner deux grands nombres premiers p et q.

-Calcul du module de chiffrement :

Calculer n=p×q.

n est le module utilisé dans les calculs de chiffrement et déchiffrement.

-Calcul de l’indicatrice d’Euler :

φ(n)=(p−1)×(q−1).

Cette valeur est utilisée pour générer les clés.

-Choix de l’exposant de chiffrement e :

Sélectionner un entier e tel que e soit premier avec φ(n) (c’est-à-dire pgcd(e,φ(n))=1) et 1≤e<φ(n).

-Calcul de l’exposant de déchiffrement d :

Trouver d tel que e×d ≡ 1 modφ(n).

Ce calcul se fait à l’aide de l’algorithme d’Euclide étendu.

-Publication et conservation des clés :

La clé publique est le couple (e,n), publié dans un annuaire public.

La clé privée est le couple (d,n), conservé secrètement par l’entité.

* + - * 1. **Chiffrement d’un message :**

L’expéditeur utilise la clé publique du destinataire pour chiffrer un message M.

-Transformation du message :

Convertir le message M en un entier M (par exemple, en utilisant le code ASCII).

Vérifier que M<n.

-Calcul du message chiffré C :

C=Me mod n.

Le message chiffré C est envoyé au destinataire.

* + - * 1. **Déchiffrement du message :**

Le destinataire utilise sa clé privée pour déchiffrer le message reçu.

-Calcul du message en clair M :

M=Cd mod n.

Le théorème d’Euler garantit que cette opération restitue M sans ambiguïté.

-Reconstruction du message :

Convertir M en texte clair à l’aide de la méthode inverse de transformation.

* + - 1. **ElGamal:**

Le chiffrement ElGamal est un protocole de cryptographie asymétrique inventé par Taher Elgamal en 1984. Basé sur la difficulté du problème du logarithme discret dans les groupes cycliques multiplicatifs, il permet de sécuriser les messages grâce à l’utilisation d’une paire de clés, une publique et une privée, et un générateur. Ce système repose sur le fait que le calcul inverse de x à partir de gx mod p est computationnellement difficile, garantissant ainsi la sécurité.

Fonctionnement :

* + - * 1. **Génération des clés :**

Choisir un grand nombre premier p.

Sélectionner deux entiers a et g tels que a<p et g<p.

Calculer A=ga mod p.

La clé publique est composée de (A,g,p), tandis que la clé privée est a.

* + - * 1. **Chiffrement :**

Pour chiffrer un message M, choisir un entier aléatoire b tel que

b<p et pgcd(b,p−1)=1.

Effectuer les calculs suivants:

-B=gb mod p (première composante du message chiffré).

-C=M⋅Ab mod p (deuxième composante du message chiffré).

Le message chiffré est constitué de la paire (B,C).

* + - * 1. **Déchiffrement :**

Pour déchiffrer un message (B,C), utiliser la clé privée a et calculer :M=C⋅(Ba)−1 mod p.

Ce calcul permet de retrouver M, car Ba mod p = gab, annulant ainsi la composante chiffrée du message.

* + 1. **Comparaison entre le chiffrement symétrique et le chiffrement asymétrique :**

|  |  |
| --- | --- |
| **Chiffrement symétrique** | **Chiffrement asymétrique** |
| -Chiffrement à clé privé (utilisation d'une clé pour crypter qui fonctionne aussi pour décrypter).  -Très facile.  -Très rapide.  -Clés courtes (128 ou 256 bits).  -les clés de chiffrement symétrique doivent être conservées en toute sécurité - vous devez vous assurer que chaque personne qui a besoin de la clé, il obtient sans aucun risque de le sortir. | -Chiffrement à clé publique (utilisation de deux clés, une pour crypter **clé publique** et autre pour décrypter **clé privée** ).  -Difficile par rapport au cryptage symétrique. -Plus lent.  -Des clés plus longues (1024 à 4096 bits).  -les clés publiques qu'ils utilisent sont sans danger pour être publié n'importe où parce que pour obtenir la clé privée à partir d'une clé publique peut prendre des centaines d'années de travail. |

* + 1. **La cryptographie hybride :**

La cryptographie hybride est un système combinant les forces des deux grandes familles de la cryptographie : la cryptographie symétrique et la cryptographie asymétrique. Ce type de cryptographie vise à exploiter les avantages des deux approches pour assurer à la fois efficacité et sécurité dans le processus de chiffrement des données.

* + - 1. **Principe de la Cryptographie Hybride:**
         1. **Efficacité et rapidité :**

La cryptographie symétrique, bien adaptée au traitement de grandes quantités de données, offre une vitesse et une efficacité remarquables. Cependant, elle présente une faiblesse majeure : la difficulté d'assurer un partage sécurisé de la clé secrète.

* + - * 1. **Sécurité dans l'échange de clés :**

La cryptographie asymétrique, bien qu’elle soit plus lente en raison de la complexité des calculs , excelle dans l’échange sécurisé de clés grâce à l’utilisation d’une clé publique et d’une clé privée.

* + - * 1. **Combinaison des deux approches :**

La cryptographie hybride tire parti des avantages des deux méthodes :

L’algorithme asymétrique est utilisé pour sécuriser la transmission de la clé de session.

L’algorithme symétrique assure un chiffrement rapide et efficace des données, garantissant des performances optimales.

* + - 1. **Exemples de Crypto-systèmes Hybrides:**
         1. **PGP (Pretty Good Privacy) :**

PGP est un système de cryptographie hybride qui combine les avantages de la cryptographie symétrique et asymétrique. Il garantit une sécurité élevée, une distribution efficace des clés, et une rapidité de traitement grâce à l'utilisation conjointe de ces deux techniques.

**Étapes du chiffrement avec PGP :**

***- Compression des données :***

- Les données en texte clair sont d'abord compressées.

- Cette étape réduit la taille du fichier, économise de l'espace de stockage, accélère la transmission, et renforce la sécurité cryptographique en minimisant les schémas exploitables dans le texte clair.

*-* ***Génération de la clé de session :***

- Une clé de session aléatoire est générée.

- Cette clé est temporaire et utilisée pour un seul message, assurant une sécurité accrue.

Elle est créée à partir de sources aléatoires, telles que les frappes de touches ou les déplacements de la souris.

***- Chiffrement des données :***

- Les données compressées sont chiffrées avec la clé de session en utilisant un algorithme de cryptage symétrique rapide et sécurisé (comme IDEA).

***- Chiffrement de la clé de session :***

- La clé de session est ensuite chiffrée à l’aide de la clé publique du destinataire en utilisant un algorithme asymétrique (comme RSA).

***- Transmission des données :***

**-** Le message chiffré et la clé de session chiffrée sont envoyés ensemble au destinataire.

**Étapes du déchiffrement avec PGP :**

***- Récupération de la clé de session :***

Le destinataire utilise sa clé privée RSA pour déchiffrer la clé de session chiffrée.

***- Déchiffrement des données :***

Avec la clé de session récupérée, le destinataire déchiffre les données chiffrées en utilisant l'algorithme symétrique (IDEA).

***- Décompression des données :***

Enfin, les données déchiffrées sont décompressées pour retrouver le texte original.

**GnuPG (GNU Privacy Guard) :** est un logiciel de cryptographie hybride conforme au standard OpenPGP. Il prend en charge les algorithmes suivants :

**Algorithmes asymétriques :** RSA, ElGamal, DSA.

**Algorithmes symétriques :** 3DES, IDEA, CAST5, Blowfish, AES, Camellia.

* + 1. **Fonction de hachage :**
       1. **Qu’est-ce qu’une fonction de hachage ?**

Une fonction de hachage, également appelée fonction de hachage à sens est une fonction mathématique qui transforme un ensemble de données de taille arbitraire en une empreinte numérique de taille fixe. Cette empreinte, souvent appelée somme de contrôle, résumé de message, condensé ou empreinte cryptographique, est utilisée pour diverses applications en cryptographie.

* + - 1. **Propriétés d'une Fonction de Hachage :**

Une bonne fonction de hachage H doit répondre aux propriétés suivantes :

* **Traitement de données de toute taille :** H peut être appliquée à des blocs de données de taille variable.
* **Sortie de longueur fixe :** La valeur produite (empreinte) a une longueur constante, quelle que soit la taille de l'entrée.
* **Facilité de calcul :** Le calcul de H(x) pour une donnée x doit être rapide et efficace.
* **Résistance à l'inversion** : Il est informatiquement impossible de retrouver x à partir de H(x).
* **Résistance à la recherche de collisions :** Trouver y ≠ x tel que H(y)=H(x) est extrêmement difficile.

**- Résistance aux collisions totales :** Il est impossible de trouver une paire (x,y) telle que H(x)=H(y).

- Une fonction de hachage qui respecte uniquement les cinq premières propriétés est considérée comme une fonction de hachage faible. Si la sixième propriété est également satisfaite, la fonction est alors qualifiée de fonction de hachage forte.

* + - 1. **L’utilisation des fonctions de hachage cryptographique :**

Les fonctions de hachage sont utilisées dans divers applications telles que :

* **Authentification des messages :** est obtenue à l'aide du code d'authentification des messages.
* **Signatures numériques :** la valeur de hachage du message est chiffrée avec la clé privée de l'utilisateur, afin que le destinataire connaisse la clé publique de l'utilisateur, le destinataire puisse vérifier l'intégrité du message.
* **Fichiers de mots de passe unidirectionnels :** dans la plupart des cas, la valeur de hachage des mots de passe est stockée au lieu du mot de passe lui-même, cela protège les mots de passe même si quelqu'un accède aux fichiers qui contiennent des mots de passe.
* **Détection de virus :** une valeur de hachage de chaque fichier est stockée en toute sécurité, plus tard, il est possible de déterminer si un fichier a été modifié ou non en recalculant simplement la valeur de hachage du fichier et en comparant la nouvelle valeur avec celle stockée en toute sécurité.
  + - 1. **Exemple de fonction de hachage :**
         1. **Algorithme SHA (Secure Hash Algorithm):**

Le Secure Hash Algorithm (SHA) est une famille de fonctions de hachage cryptographiques conçue pour garantir la sécurité et l'intégrité des données. Développée par la NSA et standardisée par le NIST, cette famille d'algorithmes est largement utilisée dans diverses applications de cryptographie telles que les signatures numériques, l'authentification et la vérification d'intégrité.

**Les Versions de SHA :**

**SHA-1 :**

- Taille de l’empreinte : 160 bits (40 caractères).

- Longtemps populaire, mais désormais considérée comme vulnérable en raison d'attaques exploitant des collisions.

- Déconseillée depuis 2010 par le NIST au profit de versions plus robustes.

**SHA-224 :**

- Version optimisée introduite pour des applications spécifiques.

- Produit une empreinte de 224 bits, offrant un bon compromis entre sécurité et performance.

**SHA-256 :**

- Fournit une empreinte de 256 bits, augmentant considérablement la résistance aux attaques par force brute.

- Recommandée pour remplacer SHA-1 dans de nombreuses applications modernes.

**SHA-384 :**

- Produit une empreinte de 384 bits, offrant une sécurité accrue.

- Cependant, plus intensif en calcul, ce qui limite son utilisation dans les systèmes à ressources limitées comme les appareils mobiles.

**SHA-512 :**

* Fournit une empreinte de 512 bits, idéale pour des applications nécessitant un haut niveau de sécurité.
* Comme SHA-384, il est intensif en calculs et mieux adapté aux systèmes puissants.

1. Introduction au BIG DATA :
   1. Définition du Big Data :

Le Big Data fait référence à l’explosion de l’ensemble de données massifs et complexes qui ne peuvent pas être traités efficacement avec les méthodes traditionnelles de traitement des données, à leur variété, et aux nouvelles solutions proposées pour gérer cette volumétrie, et qui dépassent en général les capacités d'une seule et unique machine et nécessitent des traitements parallélisés.

* 1. Type des données Big Data :

Une structure de données est un moyen qui permet d'organiser et de stocker des données, afin qu'elles puissent être consultées et modifiées efficacement.

Dans le Big Data, les données collectées, stockées et traitées peuvent être issues de différents domaines et crées par plusieurs sources, ce qui génère une masse de données des types différents structurés et non structurés, semi- structurées :

* + 1. **Données structurées** :

Cela signifie que les données sont organisées dans une structure reconnaissable, faciles à stocker et à analyser. Elle peut être recherchée avec des algorithmes de recherche simples et directs par type de données par exemple.

Une base de données relationnelle comme le langage de requête structuré (SQL) représente un bon exemple pour les données structurées, il contient des nombres organisés, des dates, des groupes de mots.

* + 1. **Données non-structurées** :

Les données non structurées sont des informations, sous de nombreuses formes différentes (comme des images / objets, du texte, des e-mails), qui ne correspondent pas généralement aux modèles de données relationnelles traditionnelles. Cela rend le traitement et l'analyse très difficiles et longues.

* + 1. **Données semi-structurées** :

Les données semi-structurées sont des données irrégulières qui peuvent être incomplètes et avoir une structure qui change rapidement mais qui ne se conforme pas à un schéma fixe ou explicite (comme des informations de plusieurs sources).

* 1. Caractéristiques de Big Data (5V du Big data) :
     1. **Volume** : Le volume se réfère à la quantité de données générées quotidiennement par des entreprises ou des personnes.
     2. **Vitesse** : La vitesse est une caractéristique importante du Big Data, elle signifie la vitesse de génération des données.
     3. **Variété** : Les données à traiter sont sous forme structurés ou non structurés ou semi-structurées.
     4. **Véracité** : La véracité fait référence à la précision et la fiabilité des données. Pour extraire la valeur des données, les données doivent être nettoyées pour supprimer le bruit, ce nettoyage des données est important pour que les données incorrectes et défectueuses puissent être filtrées.
     5. **Valeur** : La valeur des données se réfère à l'utilité des données en fonction du but prévu. L'objectif final de tout le système d'analyse des big data consiste à extraire cette valeur des données.
  2. Technologies et plateformes pour Big Data :

Pour pouvoir traiter des bases de données volumineuses, plusieurs solutions ont été proposées.

* + 1. **Hadoop :**

C’est un ensemble d'outils et d'applications qui sont utilisés pour analyser de grandes quantités et types de données. Hadoop est une approche du traitement des données qui diffère radicalement du modèle de base de données relationnelle traditionnel. Il s'agit d'un logiciel "open-source" qui fonctionne dans le réseau d'ordinateurs en parallèle pour trouver des solutions au Big Data et le traiter à l'aide de l'algorithme MapReduce.

* + 1. **Apache Spark:**

Apache Spark est un Framework de traitement distribué, open source pour les charges de travail Big Data. Il utilise la mémoire cache et la fonction d'exécution de requête optimisée pour interroger rapidement des données de toute taille.

Spark est un moteur rapide pour le traitement de données à grande échelle. Moteur Spark Core utilise l'ensemble de Données Distribué Résilient, ou RDD (Resilient Distributed Data), comme type de données de base.

Apache Spark peut effectuer rapidement des tâches de traitement sur de très grands ensembles de données, et peut également distribuer des tâches de traitement de données sur plusieurs ordinateurs.

* + 1. **MongoDB :**

C’est une base de données NoSQL open source. Puisqu'il s'agit d'une base de données non relationnelle, elle peut traiter des données structurées, semi-structurées et non structurées. Elle utilise un modèle de données non relationnel, orienté document, et un langage de requête non structuré.

MongoDB stocke les données dans des documents flexibles de type JSON, ce qui signifie que les champs peuvent varier d'un document à l'autre et que la structure des données peut être modifiée au fil du temps.

Effectuez des opérations CRUD (*create, read, update and delete document*s) dans l'interface utilisateur de MongoDB Atlas ou en utilisant l'API de requête MongoDB, avec ou sans transactions.

MongoDB fournit diverses fonctionnalités, telles que l'authentification, le contrôle d'accès, le cryptage, pour sécuriser les bases des données.

* 1. Application du Big data :
     1. **La santé** :

Le Big Data a déjà commencé à créer une énorme différence dans le secteur de la santé. Grâce à l'analyse prédictive, les professionnels de la santé sont désormais en mesure de fournir des services de soins de santé personnalisés aux patients individuels.

* + 1. **Le secteur bancaire :**

Le secteur bancaire s'appuie sur le Big Data pour la détection des fraudes. Les outils Big Data peuvent détecter efficacement les actes frauduleux en temps réel tels que l'utilisation abusive des cartes de crédit, l'archivage des pistes d'inspection ou la modification défectueuse des statistiques client.

* + 1. **L'internet des objets :**

C’est l'un des plus grands utilisateurs de Big Data, les entreprises informatiques du monde entier utilisent le Big Data pour optimiser leur fonctionnement, améliorer la productivité des employés et minimiser les risques dans les opérations commerciales. En combinant les technologies Big Data avec le domaine du Machine Learning (ML) et l’intelligence Artificielle (IA), le secteur informatique propulse continuellement l'innovation pour trouver des solutions même pour les problèmes les plus complexes.

* + 1. **Personnalisation de l'expérience client :**

Personnaliser le contenu, la présentation et l'expérience utilisateur en fonction des préférences individuelles. Par exemple, afficher des recommandations de produits personnalisées, ajuster la page d'accueil en fonction de l'historique de navigation ou personnaliser le menu de navigation.

1. Les risques de la Sécurité dans le Big Data :

La sécurité joue un rôle essentiel dans la communication de données sensibles via le canal de communication non sécurisé. Les attaques de sécurité d’information ou du réseau sont classées comme des attaques passives et actives.

* 1. Attaques passives :

La motivation derrière les attaques passives est de surveiller les services et la communication de deux parties engagées dans la communication (émetteur et récepteur).

Deux types d’attaques passives sont classés, à savoir la publication du contenu des messages et l’analyse du trafic. La publication du contenu d’un message est un type d’attaque qui analyse et lit le message délivré entre l’émetteur et le récepteur. Dans le cadre de l’analyse du trafic, les modèles de transmission de données sont étudiés et tentent d’extraire les données cachées d’origine.

* 1. **Attaques actives :**

L’attaquant est directement impliqué dans l’attaque en faisant son impression sur la communication du réseau. Cette attaque comprend la modification des données, le piratage des ressources et la fausse relecture des messages, le retard et le déni des services.

* **Masquage** :

L’intrus obtient l’accès non autorisé en obtenant l’autorisation autorisée d’accéder aux informations confidentielles.

* **Piratage des ressources** :

Les pirates informatiques obtiennent l’accès non autorisé aux ressources des deux parties engagées dans la transmission de données confidentielles. Les ressources peuvent être un ordinateur, un réseau ou un périphérique de mémoire.

* **Capture non autorisée d’informations** :

Les données circulent d’une partie à une autre, obtiennent un accès non autorisé, silencieux et sans informer les propriétaires d’origine.

* **Modifications non autorisées des informations :**

Le blocage du service d’origine, l’accès aux informations, la modification du message d’origine et son renvoi.

* **Déni ou répudiation de service** :

Le trafic de l’émetteur est complètement arrêté en régulant les ressources de communication et en donnant les faux messages en bloquant les services.

* 1. Quelques solutions de sécurité pour le Big Data :

Ces solutions de chiffrement et de sécurité offrent des mécanismes robustes pour protéger les données dans divers environnements de bases de données, assurant ainsi la confidentialité et l'intégrité des informations sensibles.

* + 1. **Chiffrement NoSQL :**

Les bases de données NoSQL, telles que Cassandra, Couchbase et Redis, offrent également des solutions de chiffrement :

* **Cassandra** : Utilise le chiffrement au repos avec des clés gérées par Key Management Service et supporte également le chiffrement en transit avec TLS.
* **Chiffrement MongoDB**: MongoDB propose plusieurs fonctionnalités de sécurité.

Et il y a d’autres systèmes de gestion de bases de données comme (Chiffrement de Microsoft SQL Server, Chiffrement base de données Oracle, Chiffrement base de données PostgreSQL…) pour les données massives, structurées, semi-structurées.

* 1. Défis de la Cryptographie dans le Big Data :
* ***Scalabilité :*** Adapter les méthodes cryptographiques pour gérer de grands volumes de données.
* ***Performance :*** Maintenir un traitement rapide et efficace malgré l'ajout de la sécurité.
* ***Complexité :*** Implémentation de solutions cryptographiques adaptées aux divers types de données et aux architectures distribuées.

1. Techniques de cryptographie adaptés au big data :
   1. Chiffrement homomorphe :

C’est une technique de cryptographie correspond à des systèmes de cryptographie asymétrique, permettant de réaliser des opérations sur des données chiffrées tel que le résultat déchiffré d’un traitement effectué sur ces données est identique à celui obtenu à partir de la même opération effectuée sur des données non chiffrées (Le résultat de ces opérations reste chiffré et ne peut être déchiffré que par les destinataires autorisés), Cette technique permet aux participants de garder leurs données confidentielles.

En d'autres termes, il permet d'exécuter des opérations mathématiques directement sur les données chiffrées, et le résultat de ces opérations, une fois déchiffré, correspondra au résultat des mêmes opérations effectuées sur les données en clair.

* + 1. **Types de chiffrement homomorphe :**

Il existe plusieurs formes de chiffrement homomorphe, suivant si l’algorithme peut gérer la commutativité de certaines opérations mathématiques.

* + - 1. **Chiffrement partiellement homomorphe (PHE) :**

Permet une seule opération mathématique (addition ou multiplication) sur les données chiffrées, comme RSA (multiplication).

* + - 1. **Chiffrement homomorphe à seuil (SHE) :**

Permet un nombre limité d'opérations mathématiques (généralement les additions et un nombre limité de multiplications), comme BFV (Brakerski-Fan-Vercauteren).

* + - 1. **Chiffrement totalement homomorphe (FHE) :**

Permet un nombre illimité d'additions et de multiplications sur les données chiffrées. C'est le type le plus puissant et le plus flexible, mais aussi le plus complexe à mettre en œuvre.

* + 1. **Principe de base du chiffrement homomorphe :**
       1. **Chiffrement** :

Les données en clair **m** sont chiffrées à l'aide d'une clé publique pour produire des données chiffrées**.**

* + - 1. **Calcul sur les données chiffrées** :

Les opérations mathématiques (comme l'addition et la multiplication) sont effectuées directement sur **c** sans le déchiffrer **c** .

* + - 1. **Déchiffrement** :

Le résultat chiffré des opérations est ensuite déchiffré à l'aide de la clé privée pour obtenir le résultat en clair.

Formellement, pour un schéma homomorphe Enc et des opérations + et \*, ona :

* + 1. **Cas d’usage du chiffrement homomorphe :**
       1. **Analyse de Big Data :**

Les entreprises peuvent effectuer des analyses complexes sur de grandes quantités de données chiffrées, extrayant les informations précieuses tout en garantissant la confidentialité des données brutes. Cela est important dans des domaines comme le marketing, où les données des clients doivent être protégées.

* + - 1. **Systèmes de vote électronique :**

Le chiffrement homomorphe permet de chiffrer les votes des électeurs de manière à ce qu'ils puissent être comptés sans être déchiffrés, assurant ainsi à la fois la confidentialité du vote et l'intégrité du processus de comptage. Cela est essentiel pour garantir des élections sécurisées et transparentes.

* + - 1. **Sécurisation des IoT (Internet of Things) :**

Les appareils IoT collectent et transmettent souvent des données sensibles. Le chiffrement homomorphe peut sécuriser ces données pendant leur transmission et leur traitement, garantissant ainsi la confidentialité et l'intégrité des informations collectées par ces dispositifs.

* 1. Chiffrement par attribut :

**Le chiffrement par attributs** est une forme de chiffrement à clé publique un-à-plusieurs où une seule clé publique est utilisée pour chiffrer les données, mais la possibilité est offerte de générer plusieurs clés privées pour déchiffrer ces données. Ces clés privées sont générées en fonction des attributs des utilisateurs.

Le chiffrement par attributs ABE a été introduit en 2005 par Sahai et Waters comme une évolution du chiffrement basé sur les identités floues, lui-même étant une amélioration du chiffrement basé sur l'identité(IBE).

* + 1. **Les variantes principales du chiffrement par attributs(ABE) :**

Dans l’ABE, les données sont chiffrées et déchiffrées en fonction d’attributs et de la politique d'accès aux données. Seules les entités avec des attributs qui satisfont une politique d'accès aux données peuvent déchiffrer un texte. Les deux principales variantes sont CP-ABE(Ciphertext Policy Attribute Based Encryption) et KP-ABE(Key Policy Attribute Based Encryption).

* + - 1. **La variante KP-AB :**

Key-Policy Attribute Based Encryption(KP-ABE) qui a été développée par Goyal et al en 2006. Pour KP-ABE, la politique d'accès est intégrée dans la clé secrète,en d’autres termes, on décide pour chaque utilisateur quels sont les objets auxquels il aura accès. On attache à chaque texte chiffré un ensemble d’attributs.

Une clé secrète donnée, avec une politique d'accès donnée, ne peut déchiffrer que le texte chiffré ayant les attributs qui satisfont sa politique d'accès.

***Exemple :***

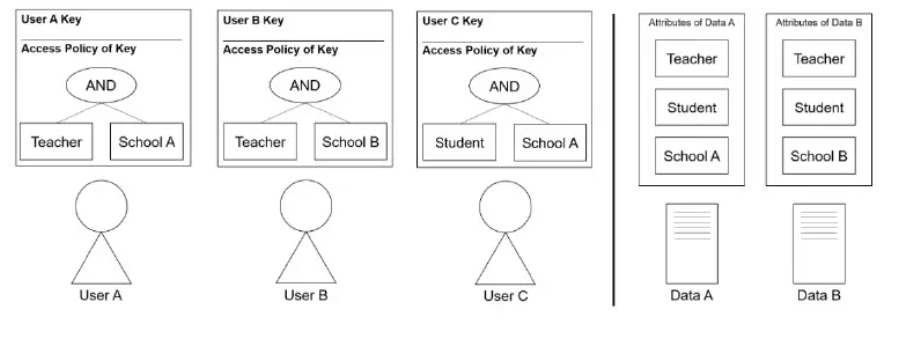


Figure:KP-ABE

* + - 1. **La variante CP-AB :**

Ciphertext-Policy Attribute Based Encryption(CP-ABE) est proposée pour la première fois par Bethencourt et al en 2007, dans laquelle la politique d'accès est intégrée dans le texte chiffré et les clés secrètes sont générées avec un ensemble d’attributs décrivant l’utilisateur légitime qui pourra déchiffrer ce texte. Seul les clés secrètes avec un ensemble d’attributs qui satisfait

La politique d'accès peut récupérer le texte en clair.

***Exemple :***

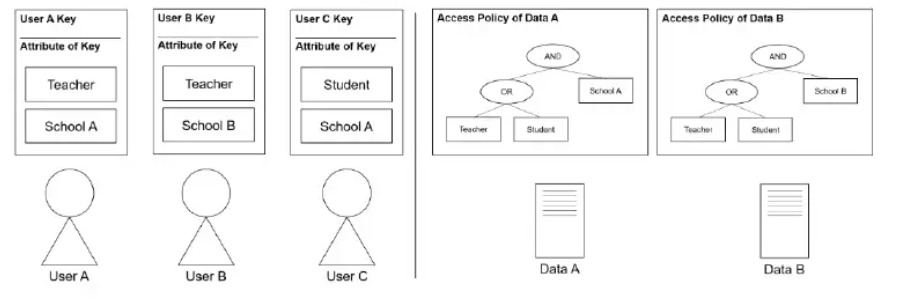


Figure:KP-ABE

* + - 1. **La comparaison entre ABE, KP-ABE et CP-ABE :**

|  | **ABE** | **KP-ABE** | **CP-ABE** |
| --- | --- | --- | --- |
| **Efficacité** | Moyen | Moyen | Elevé |
| **Résistance aux collision** | Moyen | bien | bien |
| **Surcharge de calcul** | Elevé | coût supplémentaire en termes de calcul | coût supplémentaire en termes de calcul |
| **Accès aux contrôle des grains fins** | Faible | Faible | Moyen |
| **Accès aux contrôle** | Elevé | Elevé | Elevé |

Figure : Tableau de comparaison

* 1. Fonctionnement de chiffrement par attribut :
     + 1. **Concept de base :**

Dans le chiffrement par attribut :

* Les attributs sont des propriétés qui décrivent l'utilisateur (exemples : rôle, organisation, localisation, etc.).
* Une politique d'accès est définie pour déterminer qui peut accéder aux données chiffrées.
* Deux variantes principales existent :
  + KP-ABE (Key-Policy Attribute-Based Encryption) : La politique est intégrée dans la clé privée.
  + CP-ABE (Ciphertext-Policy Attribute-Based Encryption) : La politique est intégrée dans le texte chiffré.
    - 1. **Étapes de fonctionnement :**

#### a. Phase de configuration(Setup) :

* Une **autorité centrale** génère les éléments suivants :
  + Clé maîtresse (*Master Key*) : Une clé secrète utilisée pour dériver les clés privées des utilisateurs.
  + Paramètres publics : Partagés avec tout le monde pour permettre le chiffrement.

#### b. Chiffrement des données(Encrypt) :

Un émetteur, qui souhaite partager des données sécurisées, suit ces étapes :

1. Définition de la politique d'accès :
   * Une politique d'accès est définie en termes d'attributs. Par exemple :

*(Département = Finance) ET (Rôle = Manager)*.

1. Chiffrement :
   * Les données sont chiffrées avec une clé publique et une politique d'accès associée.
   * Le texte chiffré contient les données sécurisées et la politique.

#### c. Génération des clés privées(KeyGen) :

Pour chaque utilisateur, l'autorité centrale génère une clé privée en fonction de leurs attributs. Par exemple :

* Un utilisateur ayant les attributs *(Département = Finance)* et *(Rôle = Manager)* reçoit une clé privée contenant ces propriétés.

#### d. Déchiffrement des données(Decrypt) :

Lorsqu'un utilisateur tente de déchiffrer le texte :

1. Les attributs de sa clé privée sont comparés à la politique d'accès du texte chiffré.
2. Deux scénarios possibles :
   * Si les attributs de l'utilisateur satisfont la politique : Le texte peut être déchiffré.
   * Sinon : L'accès est refusé.
     1. **Cas d’usage du chiffrement par attribut :**
        1. **Analyse de Big Data :**

Dans l’analyse de Big Data, les données sont souvent :

* Stockées sur le cloud pour une évolutivité et un accès distribué.
* Partagées entre multiples parties prenantes comme des analystes, des chercheurs ou des institutions.
* Sujettes à des réglementations strictes, par exemple :
  + Le RGPD (Règlement Général sur la Protection des Données) en Europe.
  + Les lois HIPAA pour les données médicales aux États-Unis.

Le **chiffrement par attribut** permet de gérer efficacement l’accès aux données en fonction des rôles, des responsabilités et des autorisations des utilisateurs.

#### Partage sécurisé des données dans des environnements multi-utilisateurs :

Dans des environnements où différents utilisateurs (clients, analystes, partenaires) accèdent à des sous-ensembles des données :

* Les données sont chiffrées avec des politiques d’accès basées sur des attributs comme :
  + Le rôle de l’utilisateur (*Analyste*, *Directeur*).
  + La localisation géographique (*Europe*, *Asie*).
  + Le projet ou domaine concerné (*Marketing*, *Recherche*).

#### Protection des données sensibles dans la santé (Health Big Data) :

Les données médicales des patients sont extrêmement sensibles et doivent être protégées contre les accès non autorisés :

* Les fichiers médicaux (scans, diagnostics, prescriptions) sont chiffrés en fonction des attributs du personnel médical.
* Une politique d'accès pourrait être :  
  *(Profession = Médecin) ET (Service = Cardiologie)*.

#### Analyse de données financières :

Les institutions financières utilisent le Big Data pour détecter les fraudes, analyser les tendances du marché, et évaluer les risques.

* Les données sont chiffrées en fonction de la hiérarchie et des besoins de confidentialité :
  + *(Rôle = Analyste des Fraudes)* pour accéder aux transactions suspectes.
  + *(Rôle = Directeur des Risques)* pour accéder aux prévisions économiques.

### **Avantages dans le Big Data :**

* Accès granulaire : Les données sont accessibles uniquement à ceux qui possèdent les attributs appropriés.
* Conformité réglementaire : Aide à respecter les lois sur la confidentialité en protégeant les données sensibles.
* Sécurité dans les environnements distribués : Permet un partage sécurisé des données sur des réseaux distribués et multi-tenant (par exemple, dans le cloud).
* Réduction des risques : Élimine les accès non autorisés, réduisant ainsi les risques de fuite de données.

### **Limites potentielles et solutions**

#### *Révocation des droits :*

* Révoquer un attribut ou interdire l’accès d’un utilisateur peut être compliqué.  
  **Solution** : Utiliser des techniques comme le *Chiffrement par Attribut avec Révocation* ou des mécanismes de mises à jour dynamiques.

#### *Complexité des politiques :*

* Les politiques d’accès complexes augmentent les coûts de calcul.  
  **Solution** : Optimiser les politiques pour équilibrer sécurité et performance.

#### *Surcharge de calcul :*

* Le chiffrement et le déchiffrement dans des ensembles massifs de données peuvent ralentir les performances.  
  **Solution** : Intégrer des accélérateurs matériels (par ex., GPU) ou des algorithmes de chiffrement plus efficaces.
  1. Algorithmes de chiffrement légers :

Les algorithmes de chiffrement léger sont conçus pour être efficaces en termes de consommation de ressources (comme la mémoire, la puissance de calcul et l'énergie), ce qui les rend adaptés aux environnements contraints, tels que les appareils IoT (Internet des objets), les capteurs, et les dispositifs embarqués qui ont des capacités limitées. Ces algorithmes sont également importants dans les contextes où des performances rapides sont nécessaires, tout en garantissant une sécurité adéquate.

* + 1. **Caractéristiques des algorithmes de chiffrement léger :**
* Efficacité énergétique : Ces algorithmes sont optimisés pour les appareils à faible consommation d'énergie.
* Optimisation des ressources : Utilisent moins de mémoire et de cycles CPU que les algorithmes traditionnels.
* Vitesse d'exécution rapide : Capables de chiffrer et de déchiffrer les données rapidement, ce qui est crucial dans des environnements contraints.
* Sécurité : Maintiennent un niveau de sécurité acceptable, malgré leur faible coût en ressources.
  + 1. **Exemples d'algorithmes de chiffrement léger :**

#### SPECK (Symmetric Privacy-Enhanced Communication) :

#### Est une famille de chiffrements symétriques développée par la NSA, particulièrement conçue pour les dispositifs à faible consommation.

* **Fonctionnement de SPECK** : SPECK est un algorithme de chiffrement **par bloc**, ce qui signifie qu'il chiffre des blocs de données de taille fixe en utilisant une clé secrète. Les tailles des blocs et des clés dans SPECK sont flexibles, ce qui permet de choisir la configuration optimale en fonction des contraintes du dispositif.
* Chiffrement à clé secrète : La même clé est utilisée pour le chiffrement et le déchiffrement des données, ce qui caractérise un chiffrement symétrique.
* Structure de Feistel : L'algorithme utilise une structure de type Feistel, qui divise le bloc de données en deux moitiés et applique une série d'opérations sur chaque moitié à chaque tour de chiffrement. Cela permet de garantir que chaque bit du bloc de données affecte l'ensemble du bloc, améliorant ainsi la sécurité.

- Opérations simples : Les opérations utilisées dans SPECK sont simples et peu coûteuses en termes de calculs, telles que des additions, des décalages et des XOR. Cela rend l'algorithme rapide et efficace, même pour des dispositifs avec des ressources limitées.

* + - 1. **SIMON :**

Est un autre algorithme de chiffrement symétrique léger développé par la NSA. Il est conçu pour être performant avec des tailles de blocs variables (32, 64, 128 bits) et des tailles de clés variables (de 64 à 256 bits).

* **Fonctionnement de SIMON :** SIMON utilise un chiffrement par blocs, où les données sont divisées en blocs de taille fixe, qui sont ensuite chiffrés avec une clé secrète. L'algorithme fonctionne en utilisant une structure similaire à celle de Feistel, ce qui le rend particulièrement adapté aux environnements à faible consommation d'énergie et à faible coût en termes de calcul.
* Taille des blocs : SIMON peut être configuré pour utiliser des tailles de blocs variables, comme 32 bits, 64 bits et 128 bits.
* Taille des clés : SIMON permet également des tailles de clés variables, allant de 64 à 256 bits, ce qui donne aux développeurs une grande flexibilité dans le choix de la taille de la clé en fonction des exigences de sécurité.
* Structure de Feistel : Comme SPECK, SIMON utilise une structure de type Feistel, ce qui signifie que chaque tour de chiffrement applique des transformations sur une moitié du bloc de données, avec des opérations simples comme des XOR, des décalages et des additions, garantissant une bonne diffusion et confusion des données.
* Simplicité des opérations : SIMON repose sur des opérations simples et rapides (XOR, décalages, additions), ce qui permet une implémentation efficace sur des processeurs simples ou des architectures à faible consommation d'énergie.
  + - 1. **XTEA (Extended Tiny Encryption Algorithm):**

Est une version améliorée du Tiny Encryption Algorithm (TEA). C'est un algorithme de chiffrement de bloc qui est extrêmement simple et efficace, souvent utilisé dans des systèmes à ressources limitées.

### **Fonctionnement de XTEA :** est un algorithme de chiffrement symétrique utilisant une clé secrète partagée pour le chiffrement et le déchiffrement des données. Il chiffre des blocs de 64 bits à l'aide d'une clé de 128 bits. L'algorithme utilise une structure Feistel, ce qui signifie que les données sont divisées en deux moitiés, et chaque tour de chiffrement applique des transformations sur une moitié du bloc de données.

* Opérations simples : XTEA utilise des opérations de base simples telles que des additions et des XOR (ou opérations logiques sur les bits), qui sont faciles à implémenter et à exécuter rapidement sur des processeurs simples ou à faible puissance.
* Nombre de tours : L'algorithme utilise un nombre fixe de 64 tours pour le chiffrement, ce qui est une amélioration par rapport à TEA, qui utilise moins de tours. L'augmentation du nombre de tours permet de mieux mélanger les données et d'améliorer la sécurité.
* Structure Feistel : Comme pour le chiffrement par Feistel, XTEA divise les données en deux moitiés et applique des opérations sur ces moitiés de manière répétée pendant les tours de chiffrement.
  + - 1. **Chiffrement à clé elliptique (ECC) :**

Est une méthode de cryptographie asymétrique qui repose sur les propriétés des courbes elliptiques pour créer des clés cryptographiques. Par rapport aux systèmes classiques comme RSA, le chiffrement elliptique offre un niveau de sécurité similaire avec des clés beaucoup plus petites, ce qui le rend particulièrement adapté aux environnements à ressources limitées comme les dispositifs IoT (Internet of Things), les systèmes mobiles, et autres applications à faible consommation d'énergie.

* **Fonctionnement du chiffrement à clé elliptique (ECC) :** Le chiffrement à clé elliptique repose sur le concept des courbes elliptiques, qui sont des équations mathématiques définissant des courbes sur un plan cartésien. Les propriétés de ces courbes permettent de générer des clés publiques et privées, qui sont utilisées pour effectuer des opérations cryptographiques comme le chiffrement, le déchiffrement, ainsi que la signature et la vérification des signatures.

#### Les étapes clés du processus ECC :

* Clé privée : Une clé privée est un nombre aléatoire choisi par l'utilisateur. Cette clé reste secrète et n'est pas partagée.
* Clé publique : La clé publique est générée en multipliant la clé privée par un point de base spécifique sur la courbe elliptique. La clé publique peut être partagée librement, car il est pratiquement impossible de retrouver la clé privée à partir de la clé publique grâce à la difficulté du problème du logarithme discret elliptique.
* Chiffrement et déchiffrement : Comme pour d'autres systèmes de cryptographie asymétrique, ECC utilise la clé publique pour chiffrer les messages et la clé privée pour les déchiffrer. Cependant, ECC peut accomplir cette tâche avec des clés beaucoup plus petites et des calculs plus rapides.
* Signatures numériques : ECC est également largement utilisé pour générer et vérifier des signatures numériques. Ces signatures permettent de prouver l'authenticité d'un message ou d'un document tout en préservant la confidentialité et l'intégrité des données.
  + - 1. **LEA (Lightweight Encryption Algorithm) :**

Le LEA (Lightweight Encryption Algorithm) est un algorithme de chiffrement symétrique développé pour répondre aux besoins spécifiques des applications dans des environnements à ressources limitées, tels que les dispositifs mobiles et l'Internet des objets (IoT). Il a été proposé par le gouvernement sud-coréen en 2013 comme une alternative efficace aux algorithmes de chiffrement plus lourds, comme AES, qui sont souvent trop coûteux en termes de consommation de ressources pour ces systèmes.

* **Fonctionnement de LEA** : LEA est basé sur un algorithme de chiffrement par blocs similaire à AES, mais il a été conçu pour être léger et adapté aux environnements contraints. Il prend en charge des blocs de 128 bits et des tailles de clé de 128, 192 et 256 bits, permettant un compromis entre la sécurité et les performances.

#### Caractéristiques de LEA :

* **Taille de bloc :** LEA utilise des blocs de 128 bits, ce qui est standard dans de nombreux algorithmes de chiffrement modernes, y compris AES.
* **Taille de clé :** Le système permet d'utiliser des clés de 128 bits, 192 bits, ou 256 bits, offrant une certaine flexibilité en matière de sécurité.
* **Tours de chiffrement :** Comme AES, LEA repose sur une structure de chiffrement par tours. Le nombre de tours varie en fonction de la taille de la clé :
  + 10 tours pour une clé de 128 bits
  + 12 tours pour une clé de 192 bits
  + 14 tours pour une clé de 256 bits
    1. **Cas d’usage Les algorithmes de chiffrement léger :**
       1. **Analyse de Big Data :**

Les algorithmes de chiffrement léger sont utilisés pour sécuriser les données dans des environnements où les ressources sont limitées, comme les appareils IoT (Internet of Things), les capteurs, ou les systèmes embarqués. Leur application dans l'analyse de Big Data peut être vue sous plusieurs cas d’usage. Voici quelques exemples :

### **1. Protection de la confidentialité des données dans les systèmes IoT** : Les appareils IoT génèrent d'énormes quantités de données qui peuvent être sensibles. L'utilisation d'algorithmes de chiffrement léger permet de garantir que ces données restent confidentielles même en cas de vol ou d'accès non autorisé. Ces algorithmes doivent être assez efficaces pour fonctionner sur des appareils avec des capacités de calcul limitées, tout en offrant une sécurité suffisante.

### **2. Confidentialité dans les analyses distribuées :** Les données sont souvent stockées et traitées de manière distribuée dans le Big Data, par exemple, via des clusters de serveurs. L'utilisation de chiffrement léger peut être appliquée pour garantir que même si une partie du réseau est compromise, les données restent protégées.

### **3. Protection de la vie privée dans les systèmes de traitement de données personnelles :**

### Les algorithmes de chiffrement léger peuvent être utilisés pour protéger les informations personnelles dans les analyses de Big Data, comme celles des utilisateurs dans des applications mobiles ou des réseaux sociaux. Le chiffrement permet de préserver la confidentialité des informations sensibles tout en permettant l'analyse des tendances à partir de données chiffrées.

### **4. Chiffrement des données dans les bases de données de Big Data :** Les données dans les systèmes de gestion de bases de données de Big Data, comme Hadoop ou NoSQL, peuvent être chiffrées pour prévenir l'accès non autorisé. Les algorithmes de chiffrement léger sont utilisés pour s'assurer que les données restent sécurisées sans sacrifier la performance de traitement de volumes massifs de données.

### **5. Sécurisation de l'apprentissage automatique (Machine Learning):** Lorsqu'on applique des algorithmes de machine learning sur de grandes quantités de données sensibles, le chiffrement léger peut être utilisé pour protéger la confidentialité des données tout en permettant l'analyse. Par exemple, les données peuvent être chiffrées avant d'être envoyées à un modèle d'apprentissage, ce qui assure que l'information sensible n'est pas exposée durant le processus.

### **6. Gestion sécurisée des clés dans les systèmes de Big Data :** Dans des systèmes de Big Data, la gestion des clés de chiffrement est cruciale pour assurer la sécurité des données. Les algorithmes de chiffrement léger peuvent être utilisés pour générer et gérer ces clés de manière efficace, même dans des environnements à ressources limitées.

### **7. Analyse de données chiffrées :** Une autre application dans le cadre de Big Data est l'analyse directe de données chiffrées, sans avoir besoin de déchiffrer ces données. Cela permet d'effectuer des calculs ou des analyses sur des données sensibles tout en maintenant leur confidentialité, par exemple, via des techniques comme le chiffrement homomorphique.

1. Distribution des bases de données :
   1. Introduction à la distribution des bases :
      1. **Définition des bases de données distribuées :**

Une base de données distribuée est essentiellement une base de données qui n'est pas limitée à un seul système, elle est répartie sur différents sites, c'est-à-dire sur plusieurs ordinateurs ou sur un réseau d'ordinateurs. Un système de base de données distribuée est situé sur différents sites qui ne partagent pas de composants physiques. Cela peut être nécessaire lorsqu'une base de données particulière doit être consultée par différents utilisateurs à l'échelle mondiale. Elle doit être gérée de telle sorte que pour les utilisateurs, elle ressemble à une base de données unique.

* + 1. **Types des bases de données distribuées :**
       1. **Base de données homogène :**

Dans une base de données homogène, tous les sites stockent la base de données de manière identique. Le système d'exploitation, le système de gestion de la base de données et les structures de données utilisées : tous sont identiques sur tous les sites. Ils sont donc faciles à gérer.

* + - 1. **Base de données hétérogène :**

Dans une base de données distribuée hétérogène, les différents sites peuvent utiliser des schémas et des logiciels différents susceptibles d’entraîner des problèmes dans le traitement des requêtes et les transactions. En outre, un site particulier peut ignorer complètement les autres sites. Des ordinateurs différents peuvent utiliser un système d'exploitation différent, une application de base de données différente. Ils peuvent même utiliser des modèles de données différents pour la base de données. Par conséquent, des traductions sont nécessaires pour que les différents sites puissent communiquer.

* + 1. **Le rôle dans le Big Data :**
       1. **Stockage des données massives :**
* La distribution permet de répartir les données sur plusieurs nœuds ou serveurs, ce qui facilite le stockage de grandes quantités de données que les systèmes traditionnels ne peuvent pas gérer efficacement.
* Chaque nœud gère une portion des données, évitant les limites de stockage d'un seul serveur.
  + - 1. **Traitement parallèle :**
* En distribuant les données, le traitement peut être effectué simultanément sur plusieurs nœuds (traitement parallèle).
* Cela améliore considérablement les performances pour des analyses complexes et des calculs en temps réel, caractéristiques essentielles du Big Data.
  + - 1. **Tolérance aux pannes :**
* Les systèmes distribués sont conçus pour répliquer les données sur plusieurs nœuds. En cas de défaillance d'un nœud, les autres peuvent compenser, garantissant la continuité des opérations.
  + - 1. **Scalabilité horizontale :**
* Les bases de données distribuées permettent une scalabilité horizontale, où l'on peut ajouter de nouveaux nœuds au système pour gérer des charges accrues de données et d'utilisateurs. Cela répond aux besoins croissants des systèmes Big Data.
  + - 1. **Proximité des données :**
* Dans un environnement global, la distribution géographique des bases permet de rapprocher les données des utilisateurs finaux, réduisant ainsi la latence et améliorant la vitesse d'accès.
  + 1. **Importance dans le contexte du Big Data :**
       1. **Efficacité dans la gestion des 3V du Big Data :**
* Volume : La distribution permet de gérer des pétaoctets voire des exaoctets de données.
* Vélocité : Les bases distribuées permettent un accès rapide et le traitement des flux de données en temps réel.
* Variété : Elles sont capables de stocker des types de données variés (structurées, semi-structurées, non structurées).
  + - 1. **Support pour les technologies Big Data :**
* Les bases distribuées forment le socle de plusieurs technologies Big Data comme Hadoop, Apache Cassandra, et MongoDB, qui s'appuient sur des modèles de données distribuées pour fonctionner efficacement.
  + - 1. **Optimisation des coûts :**
* En répartissant les données sur des infrastructures multiples, souvent basées sur des serveurs standards ou des solutions cloud, les entreprises réduisent leurs coûts par rapport à des solutions centralisées nécessitant des infrastructures coûteuses.
  + - 1. **Flexibilité et adaptabilité :**
* Les bases distribuées s'adaptent facilement aux besoins des entreprises en évolution, notamment dans des domaines comme l'e-commerce, l'IoT, et les réseaux sociaux, où le Big Data est omniprésent.
  + - 1. **Amélioration de la prise de décision :**
* Grâce à l'accès rapide et distribué aux données, les entreprises peuvent exploiter les analyses en temps réel pour prendre des décisions stratégiques plus rapides et éclairées.
  1. Les principes fondamentaux de la distribution des bases :
     1. **Architecture des bases distribuées :**

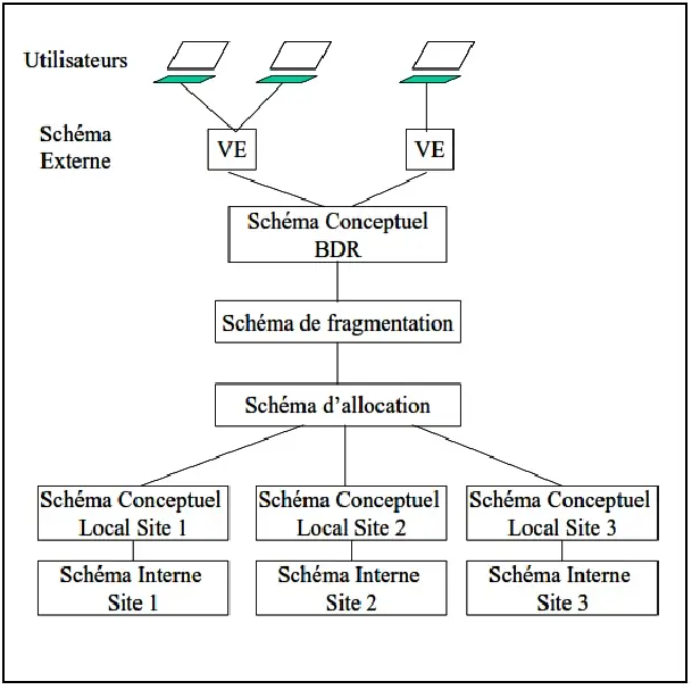
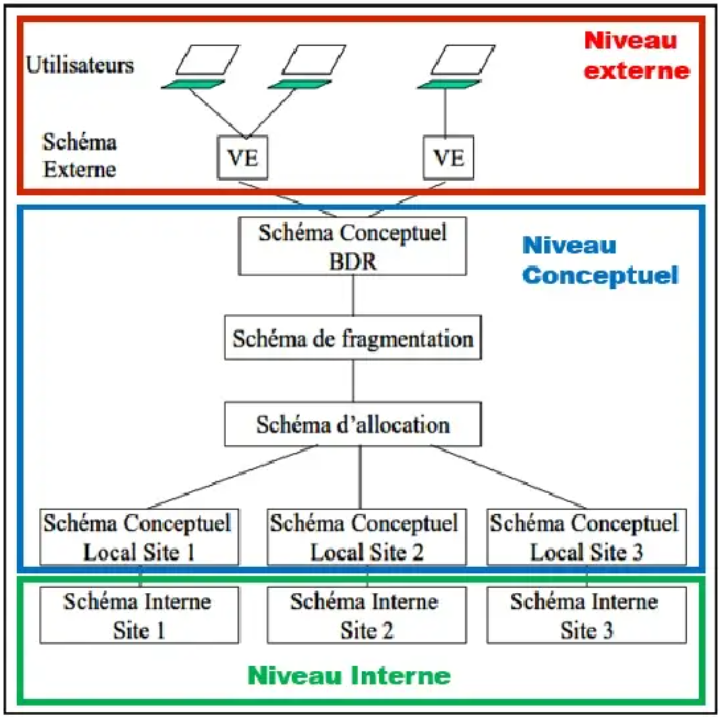
****

Figure : Architecture d’une base de données répartie

* La répartition d'une base de données intervient dans les trois niveaux de son architecture en plus de la répartition physique des données :
* **Niveau externe** : les vues sont distribuées sur les sites utilisateurs.
* **Niveau conceptuel** : le schéma conceptuel des données est associé, par l’intermédiaire du schéma de répartition (lui-même décomposé en un schéma de fragmentation et un schéma d'allocation), aux schémas locaux qui sont réparties sur plusieurs sites, les sites physiques.
* **Niveau interne** : le schéma interne global n'a pas d'existence réelle mais fait place à des schémas internes locaux répartis sur différents sites.



* + 1. **Architecture de la répartition des données :**
* Il existe deux grandes tâches dans la conception des bases de données distribuées. Premièrement, il faut établir la manière dont les nœuds communiquent entre eux. Selon l’exemple, les trois nœuds pourraient être organisés soit d’une telle manière qu’un seul nœud opère comme serveur et les autres comme clients, soit d’une manière où tous les nœuds auront une importance équivalente (figure 3).

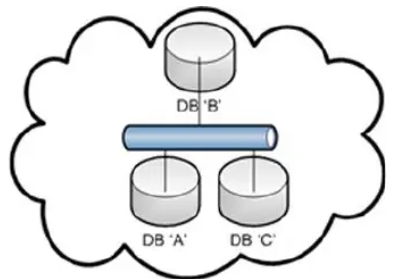


Figure 3 : Différents nœuds des bases de données distribuées

* La deuxième tâche consiste à définir la manière de répartir les données. Trois méthodes peuvent s’appliquer dans ce cas-ci : elles peuvent être partitionnées, répliquées ou hybrides.

− Dans une base de données distribuée partitionnée, la base de données entière est divisée en trois parties (A, B, C), où chacune des bases est stockée dans des nœuds différents (figure 4).

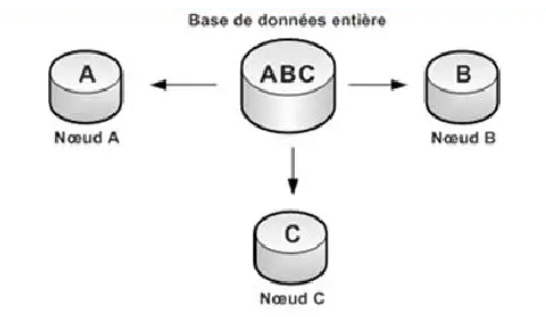
****

Figure 4 : Bases de données distribuées partitionnées.

* Dans les bases de données distribuées répliquées, la base de données entière est répliquée dans chaque nœud. Il est évident que cette méthode réduit les coûts de communication et augmente les performances du système en éliminant le besoin pour la transmission de données à des nœuds différents.

Malheureusement, cette méthode est très chère à cause de la mise à jour des données (figure 5).

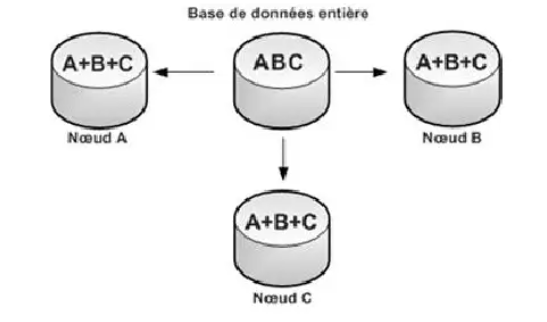


Figure 5 : Bases de données distribuées répliquées

* Les bases de données distribuées hybrides, quant à elles, sont une combinaison des méthodes précédentes. La base de données se divise sur un modèle d’utilisation, c’est -à-dire, les données sont stockées dans des nœuds où il est possible d’accéder plus fréquemment (figure 6).

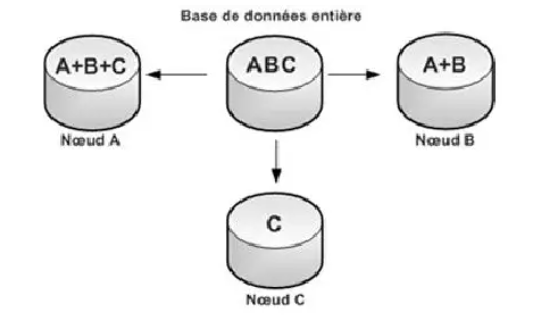
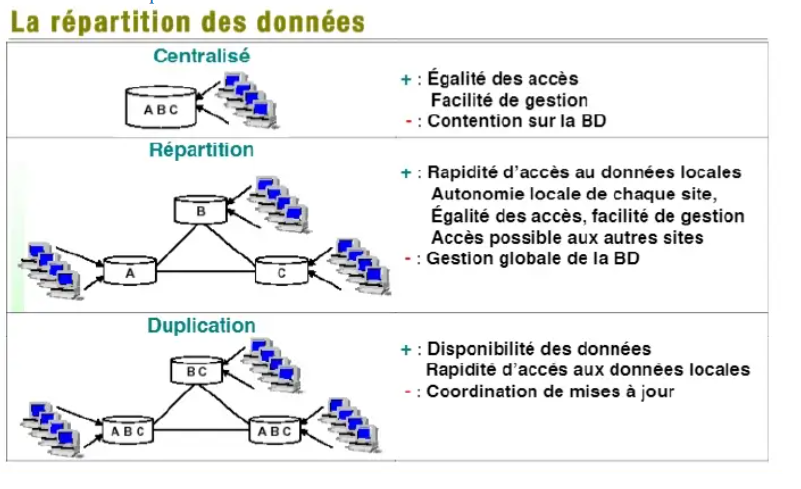


Figure 6 : Bases de données distribuées hybrides.

* + 1. **Répartition des données :**
    2. **Fragmentation des données :**
       1. **Définition :**

La fragmentation désigne le découpage de la base globale en sous bases selon les critères d'analyse.

C’est donc le processus de décomposition d'une base de données en un ensemble de sous bases de données. Cette décomposition doit être sans perte d’information. La fragmentation peut être coûteuse s'il existe des applications qui possèdent des besoins opposés. Le concepteur choisit entre un découpage horizontal, vertical ou mixte.

* + - 1. **Objectif de la fragmentation :**

Les applications ne travaillent que sur des sous-ensembles des relations. Une distribution complète des relations générerait soit beaucoup de trafic, soit une réplication des données avec tous les problèmes que cela occasionne : problèmes de mises à jour, problèmes de stockage. Il est donc préférable de mieux distribuer ces sous-ensembles.

L'utilisation de petits fragments permet de faire tourner plus de processus simultanément, ce qui entraîne une meilleure utilisation des capacités du réseau d’ordinateurs.

* + - 1. **Les problèmes de la fragmentation :**

La fragmentation peut être coûteuse s'il existe des applications qui possèdent des besoins opposés. On est en quelque sorte dans le cas d'une exclusion mutuelle qu’empêche une fragmentation correcte.

Par ailleurs, la vérification des dépendances sur différents sites peut être une opération très longue.

* + - 1. **Techniques de fragmentation :**

1. La fragmentation horizontale :

Ce type de fragmentation permet de découper une relation en sous-relations contenant des sous-ensembles des tuples de la relation mère.

La fragmentation horizontale est basée sur l’opération de sélection qui utilise un prédicat sélection prédicat prédicat (condition).

La reconstitution reconstitution reconstitution de la relation se fait par l’union l’union l’union des fragments.

Nous avons besoin de définir deux notions: prédicat prédicat prédicat simple, prédicat minterm prédicat minterm minterm.

**Prédicat simple :**