الجـــــــــــــــــــمـهـوريـــــة الجـــــــــــــزائـريـة الديمـــــــقـراطيـة الـشـــــــــعبيـة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــــــــــــــــــعـليـم العـــــــالي و البـــــــــحث العـــــلـمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــــــامعة أبي بكـــر بـلـقـايد – تـلمســـــــــــــان –

Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Télécommunications

Spécialité : Réseaux Mobiles et Services de Télécommunications

Par : Mammeri Ilhem
Guerriche Nor El Houda

Sujet

Cryptographie homomorphe pour les réseaux «Vehicular Cloud Computing»

Soutenu publiquement, le 22 / 06 /2017, devant le jury composé de :

M. Hadjila Mourad	MCB	Univ. Tlemcen	Président
M. Merzougui Rachid	MCA	Univ. Tlemcen	Examinateur
M. Moussaoui Djillali	MAA.	Univ. Tlemcen	Directeur de mémoire

Remerciement

Le premier à remercier de nous donner le pouvoir de faire ce travail est Dieu. Nous toujours cru que vous nous avez donné assez de force pour faire face à des difficultés, mais qui est non seulement ce que vous nous avez offert, tu nous as donné le bonheur et la joie tout en faisant notre travail à travers les gens autour de nous, en aidant et nous soutiennent quand et où nous avions besoin.

Nous voudrons très sincèrement remercier ***Mr Moussaoui Djilali*** *maitre de conférences classe (A) à l'Université de Tlemcen pour avoir assuré l'encadrement de ce travail. Sa disponibilité, son expérience, son savoir scientifique et ses qualités humaines ont été déterminants dans l'aboutissement de ce travail.*

Nous voudrons très sincèrement remercier ***Mr M.Hdjila*** *maitre de conférences classe (B) à l'Université de Tlemcen, d'avoir accepté de juger ce travail en présidant le jury, ainsi que* ***Mr R.Merzougui*** *maitres de conférences de classe (A) à l'université de Tlemcen ; vous nous avez honorés d'accepter de siéger parmi notre jury de mémoire.*

Pour son aide, sa compréhension et son soutien dans les moments difficiles, Nous tenons à remercier monsieur ghwali samir et son collègue monsieur walid **.**

Nous exprimons nos remerciements pour nos collègues faisant partie du groupe de notre salle pour leurs conseils et les discussions fructueuses que nous avons eu lors des réunions de groupe.

Pour terminer, nous adressons nos profonde reconnaissance à toutes celles et tous ceux que nous n'avez pas cités ici et qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

Au nom de Dieu, le clément, le très miséricordieux.

Je dédie ce modeste travail particulièrement à :

Mon papa :

J'espère qu'il trouvera dans ce travail les valeurs qu'il m'a transmis, notamment : la rigueur, la méthode, la patience et la persévérance.

Ma maman :

J'espère qu'elle trouvera aussi tout ce qu'elle m'a transmis dans le résultat de ces travaux : la générosité, la créativité et le courage d'aller jusqu'au bout.

Je n'oublie pas mon frère Mohammed , mes sœurs Chaima et Anfel

Ma grande mère

qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-elle, l'expression de mes sentiments et de mon éternelle gratitude.

A mes amies chacune son nom pour leurs aides précieuses.

A la source de mes efforts et mon bonheur, pour son précieux soutien et pour sa patience Ahmed.

A tout ceux que je n'ai pas cité leurs noms mais je n'oublie jamais leurs aides.

✍ *Ilhem.*

DEDICACES

Je dédie cette mémoire à…✍

Mes parents :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

A toute ma famille et mes amies ,

A MOHAMED .

✍ *Hoda.*

Résumé

Le chiffrement homomorphique a été introduit récemment pour subvenir aux besoins des architectures modernes utilisées dans la provision des services réseaux. La propriété fondamentale d'un système de chiffrement homomorphique est qu'il donne l'aptitude de réaliser divers traitements sur le texte chiffré sans recourir à l'opération de déchiffrement. Par exemple, de tels traitements incluent la recherche par mot-clé et la concaténation. L'avantage principal apporté par le chiffrement homomorphique est l'économie des ressources de calcul, de stockage et d'énergie. Ceci est fondamental pour plusieurs contextes tels que le Cloud Computing véhiculaire.

Ce séminaire présente les fondements mathématiques du chiffrement homomorphique et explore ses variantes (simplement homomorphique ou pleinement homomorphique). Un intérêt particulier sera accordé à l'étude de la robustesse et de la complexité des algorithmes de chiffrement homomorphique. Une revue des avantages de ce chiffrement sera réalisée en vue de souligner son adéquation aux besoins du Cloud Computing et de l'Internet des Objets.

Abstract

Homomorphic coding was introduced recently to provide for the needs for the modern architectures used in the provision of the services networks. The fundamental property of a homomorphic system of coding is that it gives the aptitude to carry out various treatments on the text quantified without resorting to the operation of deciphering. For example, such treatments include research by key word and the concatenation. The principal advantage brought by homomorphic coding is the economy of the resources of calculation, storage and energy. This is fundamental for several contexts such as vehicular Cloud Computing.

This seminar presents the mathematical bases of encryption homomorphic and explores its alternatives (simply homomorphic fully homomorphic). A private interest will be granted to the study of the robustness and the complexity of the homomorphic encryption algorithms. A review of the advantages of this coding will be carried out in order to underline its adequacy with the needs for Cloud Computing and the Internet of the Objects.

Table des matières

Chapitre II : Cloud Computing Véhiculaire

Chapitre III : Implémentation

Liste des Tableaux

Liste des Figures

Introduction générale

L'informatique en nuage, connue communément sous l'appellation «Cloud Computing » est un nouveau concept de la Technologie de l'Information qui a révolutionné ces dernières années le monde, et qui a attiré l'attention des chercheurs dans ce domaine, au lieu de déballer des ordinateurs et de les empiler dans une salle machine, le Cloud permet de télécharger virtuellement du matériel et de l'infrastructure associée à la demande.

Il se réfère à l'utilisation des capacités de stockage et de calcul des serveurs répartis dans le monde entier, liés par le réseau internet.

Les applications et les données ne se trouvent plus en local, mais dans un nuage composé d'un grand nombre de serveurs distants interconnectés au moyen d'une grande bande passante indispensable à la fluidité du système.

Le domaine du Cloud est vaste qu'il est impossible de traiter l'ensemble de ces aspects, on a le Cloud Computing véhiculaire qui est Un groupe de véhicules en grande partie autonomes dont l'informatique d'entreprise, de détection, de communication et les ressources physiques peut être coordonné et allouée dynamiquement aux utilisateurs autorisés.

dans ce travail, La sécurité du Cloud Computing véhiculaire nécessite une profonde remise en question des mesures de sécurité actuelles, partager une clé privée avec un serveur Cloud est indispensable pour qu'il puisse accéder aux données en clair et effectuer tout type de traitement sur ces derniers afin de répondre à une requête client ; cette utilisation traditionnelle de la cryptographie n'est peut-être pas la meilleure solution en termes de confidentialité et respect de la vie privée Jusqu'à présent, il était impossible de confier des données privées à un tiers et pouvoir effectuer des opérations sur ces données chiffrées sans les déchiffrer.

Pour ce faire, nous étudions les crypto systèmes homomorphes existants.

Dans ce contexte et dans le cadre de notre projet de fin d'étude, on va organiser se manuscrit de la façon suivant :

Le chapitre 1 on va donnée une définition sur la cryptographie qui définisse qu'est-ce-que la cryptologie, la cryptographie et la cryptanalyse et déférente définitions sur déférente concept de base, le cryptage homomorphique et ces algorithmes présente chacun d'eux des points positives et d'autre négatives.

Le Chapitre 2 introduit le concept du Cloud Computing, ces caractéristiques, ces modèles de déploiement et ces services après ca on va donnée une vision sur les réseaux VANET concernant architecture des réseaux sans fils ansai que le routage et leur protocole et ces application.

Le Chapitre 3 on a fait une programmation du crypto système de paillai par le langage java.

Chapitre I :

Chiffrement Homomorphe

I- 1 Introduction :

L'origine de la cryptologie mot réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : «cryptos », qui signifie caché et « logos » qui signifie mot. La cryptologie est aussi vieille que l'écriture elle-même, et a été utilisé depuis des milliers d'années pour assurer les communications militaires et diplomatiques. Par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes. Dans le domaine de l'un de cryptologie peut voir deux visions : la cryptographie et la cryptanalyse. Le cryptographe cherche des méthodes pour assurer la sûreté et la sécurité des conversations alors que le Crypto analyse tente de défaire le travail ancien en brisant ses systèmes. La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle et la cryptanalyse, à l'inverse est l'étude des procédés cryptographiques, qui dépendent d'un paramètre appelé clé. **[1]**

I-2 Définition de la cryptologie : [1]

La cryptologie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse. Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-d permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préférera le verbe chiffré. La cryptanalyse, à l'inverse, est l'étude des procédés cryptographiques dans le but de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. Le décryptement est l'action consistant à trouver le message en clair sans connaitre la clef de déchiffrement. La cryptologie, étymologiquement « la science du secret », ne peut être vraiment considérée comme une science que depuis peu de temps. Cette science englobe la cryptographie (l'écriture secrète) et la cryptanalyse (l'analyse de cette dernière).

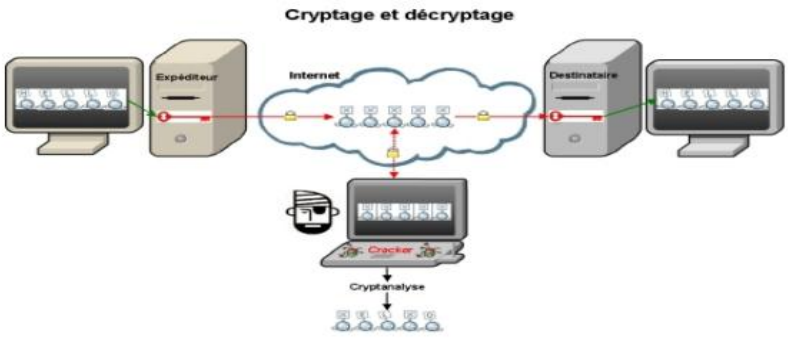


Figure 1.1 : Schéma de cryptage

I-3 Définition de la cryptographie : [1]

La cryptographie est l'art de chiffrer, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des mathématiques, de l'informatique, et parfois même de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses.

I-4 L'usage de la cryptographie : [1]

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

La confidentialité : consiste à rendre l'information intelligible à d'autres personnes que les acteurs de la transaction.

L'intégrité : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.

L'authentification : consiste à assurer l'identité d'un utilisateur, c.-à-d de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

Le non répudiation : de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

I-5 Cryptographie moderne : [1]

•Dans les années 70 et 80 la quantité des données échangées et traitées par les ordinateurs n'a pas cessé d'augmenter, ce qu'a imposé de protéger ces données.

•Généralement trois techniques de cryptographie sont généralement utilisées:

–A clef symétrique

–A clé asymétrique

–Fonction de hachage

•Ces technique ont donné lieux a de nouveau concepts comme les certificats et les signatures numériques ainsi que les infrastructures a clef publique.

I-5-1- cryptographie symétrique :

•L'un des concepts fondamentaux de la cryptographie symétrique est la clé, qui est une information devant permettre de chiffrer et de déchiffrer un message et sur laquelle peut reposer toute la sécurité de la communication.

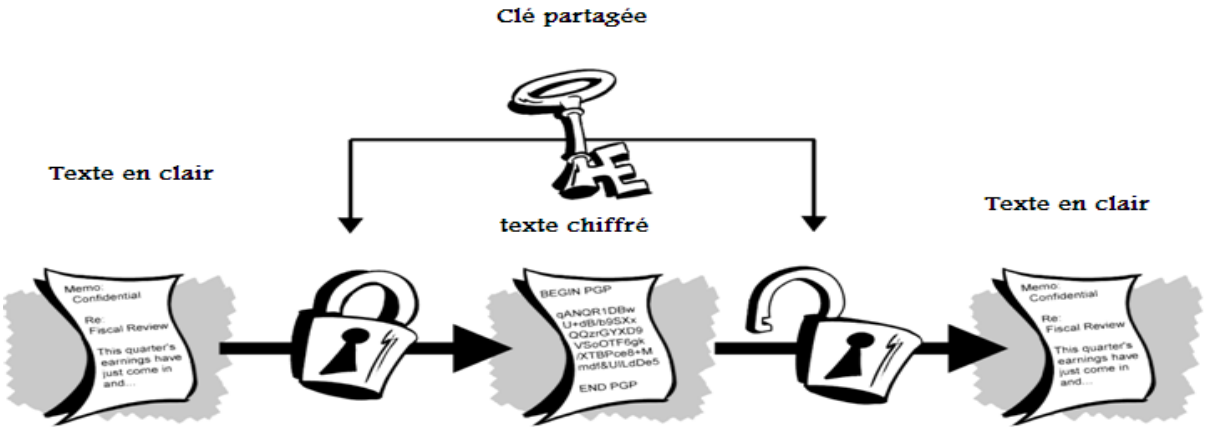
•Le problème de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sure, ce qui pose un problème dans les communications sur internet.

•Le deuxième problème de la cryptographie symétrique est que le partage de secret entre plusieurs personnes augmente le risque de son divulgation.

•Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message.

•Ces algorithmes sont dits aussi à clé secrète, une clef partagée entre les correspondants permet de chiffrer et déchiffrer les données

•Ces algorithmes ont été toujours utilisés mais en se basant sur des méthodes de substitution, la même idée a été reprise dans les 19 siècles avec plus de complexité et de rapidité en utilisant les ordinateurs.



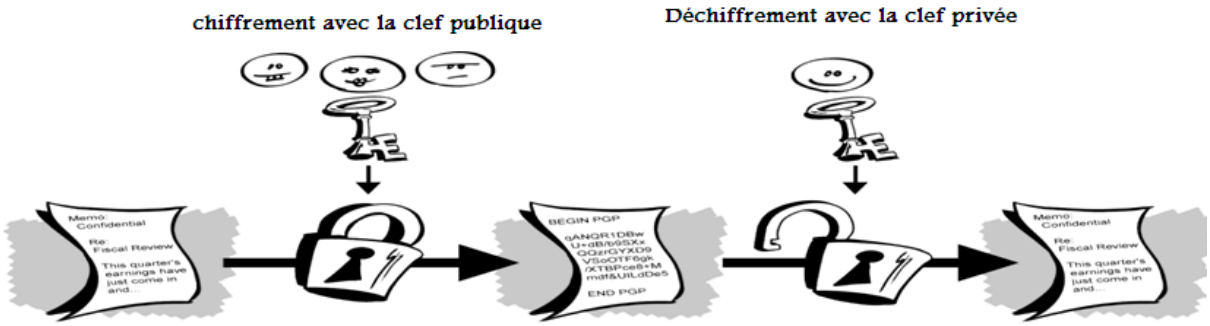
I-5-2 La cryptographie à clés asymétrique :

•L'expression «cryptographie à clé asymétrique » repose sur l'utilisation de deux clefs l'une pour le chiffrement et l'autre pour le déchiffrement

•Les deux clefs sont différentes

•Une des deux clefs est rendue publique est dite la clef publique et distribuée librement.

•La deuxième clef est dite clé privée n'est jamais distribuée et doit être gardée secrète.



•En utilisant cette paire de clés,

–Les données chiffrées par la clé publique ne peuvent être déchiffrées qu'avec la clé privée correspondante;

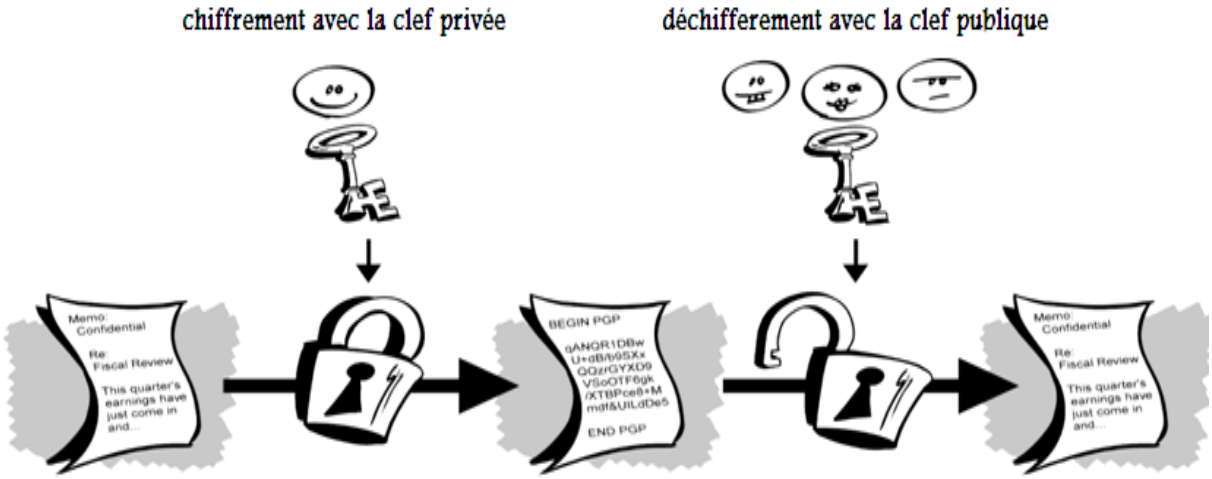
–Les données chiffrées à l'aide de la clé privée ne peuvent être déchiffrées qu'avec la clé publique correspondante.

•La cryptographie à clé publique présente un avantage majeur :

–en effet, elle permet d'échanger des messages de manière sécurisée sans aucun dispositif de sécurité.

–L'expéditeur et le destinataire n'ont plus besoin de partager des clés secrètes via une voie de transmission sécurisée.

•Cette caractéristique est utilisée pour mettre en œuvre les principes de la cryptographie et de la signature numérique.



I-5-3 Fonctions de hachage :

•Une fonction de hachage est une fonction qui fait subir une succession de traitements sur une donnée quelconque et de n'import quelle taille et produisant en sortie un empreinte de même taille quelque soit la taille des données en entrée.

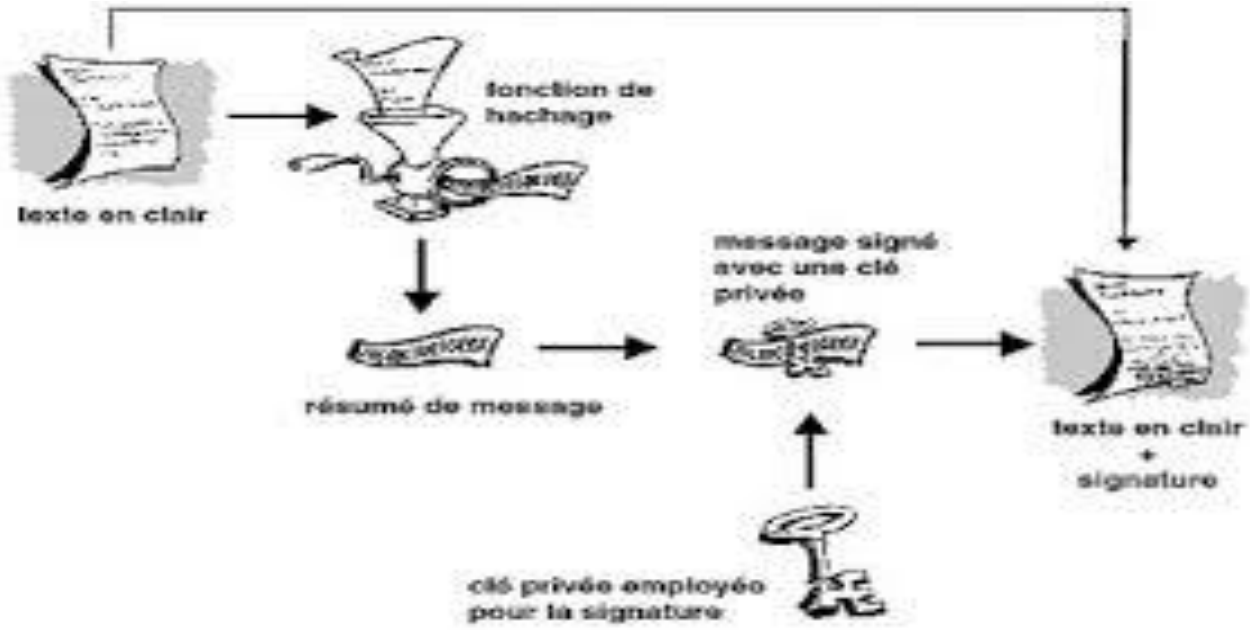
•Une fonction de hachage est une fonction à sens unique, donc le décryptage de l'empreinte produit est impossible.

•Le résultat de cette fonction est appelé somme de contrôle, résumé de message, condense, ou encore empreinte

•Les fonctions de hachage sont conçues pour effectuer un traitement de données rapide : le calcul de l'empreinte ne doit couter qu'un temps négligeable.

•Une fonction de hachage doit aussi éviter le plus possible les collisions (deux empreintes identiques alors que les données sont différentes)

•Selon l'emploi de la fonction de hachage, il peut être souhaitable qu'un petit changement de la donnée en entrée (un seul bit, par exemple) entraine une perturbation conséquente de l'empreinte correspondante.



I-6 définition de la cryptographie homomorphe : [2]

un chiffrement homomorphe est un chiffrement qui possède certaines caractéristiques algébriques qui le font commuter avec une opération mathématique, c'est-à-dire que le déchiffrement du résultat de cette opération sur des données chiffrées donne le même résultat que cette opération sur les données non chiffrées ; cette propriété permet de confier des calculs à un agent externe, sans que les données ni les résultats ne soient accessibles à cet agent.

Un exemple d'application d'un chiffrement homomorphe pour la délégation de calculs pourrait être le cas de figure où un utilisateur souhaiterait faire un calcul coûteux − dont il ne dispose pas nécessairement des ressources nécessaires pour l'exécuter − et aimerait faire appel à un service de cloud computing auquel il ne fait pas nécessairement confiance pour effectuer ses calculs.

Le chiffrement homomorphe est un domaine de recherche actif.

I-7 Historique du chiffrement Homomorphe : [2]

En 1978 Ronald Rivest, Leonard Adleman et Michael Dertouzos évoquent pour la première fois le concept de chiffrement homomorphe. Depuis, peu d'avancées ont été faites pendant 30 ans. Le système de chiffrement Shafi Goldwasser et Silvio Micali qui a été proposé en 1982 un schéma de chiffrement à sécurité prouvée. Il est homomorphe pour l'addition, mais ne peut chiffrer qu'un seul bit. Suivant le même concept, Pascal Paillier propose en 1999 un système de chiffrement efficace à sécurité prouvée qui est homomorphe pour l'addition. Peu d'années après, en 2005, Dan Boneh, Eu-Jin Goh et Kobi Nissim inventent un système de

chiffrement homomorphe à sécurité prouvée, avec lequel on peut effectuer un nombre illimité d'additions mais une seule multiplication. En 2009, Craig Gentry a proposé le premier système de chiffrement "complètement homomorphe" qui permet d'évaluer un nombre arbitraire d'additions et de multiplications et donc calculer tout type de fonctions sur des données chiffrées. Il est toujours en phase d'expérimentation car sa durée de chiffrement et de déchiffrement est loin d'être acceptable. En 2010, Van Dijk, Gentry, Halevi et Vaikuntanathan ont proposé un système de chiffrement "complètement homomorphe" avec une limitation à un seul bit.

I-8 les type de chiffrement homomorphe :

I- 8-1 Chiffrement Homomorphe additif :

Dans un contexte de chiffrement additif, un serveur distant pourra retourner le résultat d'une opération d'addition sur les messages en clair en faisant le calcul sur des messages chiffrés, sans disposer de la clé secrète.

Définition :

Un chiffrement homomorphe est additif si :

$Enc(x \otimes y) = Enc(x) \otimes Enc(y)$……………………………………………….(1)

$\prod_{i=1}^{n} Enc(m_i) = Enc(\sum_{i=1}^{n} m_i)$…………………………………….(2)

En d'autres termes, soient :

Encp une fonction de chiffrement à clé publique p.

Decs une fonction de déchiffrement à clé secrète s.

Alors :

Decs(Encp(m) * Encp(n)) = m + n…………………………….(3)

Les chiffrements qui réalisent cette propriété de chiffrement Homomorphe additif sont : Paillier et Goldwasser-Micalli.

I-8-1-1 Le chiffrement Homomorphe de Paillier : [3]

Le crypto système de Paillier est un crypto système asymétrique, conçu par Pascal Paillier en 1999. Ce crypto système est celui qui a la plus grande bande passante, appelée aussi taux d'expansion : rapport entre la longueur du clair et la longueur du chiffré. Ce crypto système est basé sur les propriétés de la fonction lambda de Carmichael dans Z.

Génération des clés :

- On choisit deux grands nombres premiers p et q ;

- On Calcule n = pq ;

On choisit un entier $g \in Z^*_{n^2}$ tel que n et L ($g^y \bmod n^2$) sont premiers entre eux, ou L désigne la fonction :

$$L : Z^*n^2 \rightarrow Z_n \quad \text{.....(4)}$$

$$u \rightarrow \frac{u-1}{n} \quad \text{.....(5)}$$

y désigne la fonction de Carmichael : y (p.q)=PPCM (p-1, q-1).

La clé publique est donc formée de (n, g) et la clé privée des deux facteurs premiers (p, q).

L'algorithme de Paillier est détaillé ci-dessous :

Crypto système de Paillier :

- **Génération des clés :**
  - Choisir p et q premiers
  - Calculer n=p.q……(7)
  - Choisir $g \in z^*_{n^2}$ tel que :
    PPCM (L ($g^y \bmod n^2$), n)=1 avec L(u)=$\frac{u-1}{n}$……(8)
    Clé publique : pk= (n, g)
    Clé privée : sk= (p, q)
- **Chiffrement : Enc (m, pk, r)**
  - Choisir $r \in Z^*_n$
  - Calculer $c= g^m.r^n \bmod n^2$

❖ **Déchiffrement : Dec(c,sk)**

- Calculer $m=\frac{L(c^y \bmod n^2)}{L(g^y \bmod n^2)} \bmod n$.....(9)

Supposons qu'on a c1 et c2 deux textes chiffrées avec l'algorithme de Paillier et m1 et m2 les textes clairs correspondants tel que :

$$C_1 = g^{m1}.r_1^{\,n} \bmod n^2 \quad \text{.....(10)}$$

$$C_2 = g^{m2}.r_2^{\,n} \bmod n^2 \quad \text{.....(11)}$$

Alors:

$$C_1.C_2 = g^{m1}.r_1^{\,n}.g^{m2}.r_2^{\,n} \bmod n^2 \quad \text{.....(12)}$$

$$= g^{m1+m2}(r_1.r_2)^n \bmod n^2 \quad \text{.....(13)}$$

Donc, le chiffrement de Pailler réalise la propriété du chiffrement Homomorphe additif.

Exemple d'application :

Soient : l'ensemble des Candidats = {X, Y, Z} et l'ensemble des Électeurs = {V1, V2, ..., Vn}

Le **tableau I.2** montre que, pour chaque candidat, on chiffre les valeurs du vote de chaque électeur avec la clé publique du chiffrement homomorphe additif "Paillier", ensuite pour obtenir la somme des votes, on déchiffre le produit des votes chiffrés comme suit :

$Dec_{sk}(\prod_{i=1}^{n} C_{i,x}) = Dec_{sk}(Enc_{pk}(Tot_x))$.....(14)

$= Tot_x$

Candidate / Electeur	X	Y	Z
V_1	$C_{1,x}=Enc(1,pk_A)$	$C_{1,y}=Enc(0,pk_A)$	$C_{1,z}=Enc(0,pk_A)$

V_2	$C_{2,x}$= Enc(1,pk_A)	$C_{2,y}$=Enc(0,pk_A)	$C_{2,z}$=Enc(0,pk_A)
V_3	$C_{3,x}$=Enc(0,pk_A)	$C_{3,y}$=Enc(0,pk_A)	$C_{3,z}$=Enc(1,pk_A)
. . .	. . .	. . .	. . .
V_N	$C_{n,x}$=Enc(0,pk_A)	$C_{n,y}$=Enc(1,pk_A)	$C_{n,z}$=Enc(0,pk_A)
Total	?	?	?

Tableau I.2 : les votes des électeurs chiffrés un par un

Le **tableau I.3** exprime le résultat obtenu après déchiffrement via la clé privée du chiffrement homomorphe de "Paillier" .

CHIFFREMENT HOMOMORPHE ADDITIF :

Candidats Electeurs	X	Y	Z
V_1	1	0	0
V_2	1	0	0
V_3	0	0	1
. . .	. . .	. . .	. . .
V_n	0	1	0
Total	Tot_x	Tot_y	Tot_z

Tableau I.3 : Application de "Paillier" pour retrouver la somme des votes

I-8-1-2 Le chiffrement Homomorphe de Goldwasser-Micalli : [4]

Le crypto système de Goldwasser-Micalli (GM) est un algorithme asymétrique de cryptographie à clé publique, développé par Shafi Goldwasser et Silvio Micali en 1982. Goldwasser et Micali ont introduit la notion de chiffrement probabiliste, tout système de chiffrement doit intégrer de l'aléa dans le processus de chiffrement pour être considéré comme sûr. Le schéma de GM qui repose sur la difficulté du problème de la résiduosité quadratique n'est pas efficace : les textes chiffrés peuvent être des centaines de fois plus longues que les textes d'origine.

Génération des clés :

Le problème de résiduosité quadratique : étant donné un entier composite impair n, et un a € Z_n^* tel que (a/n)=1 décider si (a) est ou non un résidu quadratique modulo n.

Supposons n = pq (produit de deux nombres premiers), (a/n) = 1 implique soit (a/p) = (a/q) = 1 (a est résidu quadratique) ou (a/p) = (a/q) = ≠1 (a est non-résidu quadratique).

- On Choisit p et q premiers ;

- On Calcule n=pq ;

- On Choisit z € Z_n tel que z soit un résidu non quadratique modulo n et (z /n) = 1

La clé publique est donc formée de (n, z) et la clé privée des deux facteurs premiers (p, q).

L'algorithme de Goldwasser-Micalli se présente comme suit :

Crypto système de Goldwasser-Micalli :

- **Génération des clés :**
  - Choisir p et q premiers.
  - Calculer n= p.q.
  - Choisir z∈Z_n tel que : $(\frac{z}{n})$=1 et $(\frac{z}{p})$=-1

    Clé publique : pk= (n, z)
    Clé privée : sk= (p, q)

- **Chiffrement : Enc (m_i,pk,r_i) :**
  Soit M=$\{0,1\}$ l'espace des messages en clair :
  Pour tout m∈M ,m est composé de t bit :m_1 m_2.....m_t
  - Choisir aléatoirement pour $\forall_i \in [1,t]$ un r_i
  - Calculer $c_i=Z^{mi}.r_1^2$ mod n

❖ **Déchiffrement : Dec (c_i,sk) :**

- Calculer $(\frac{c_i}{p})$=e_i pour $\forall_i \in [1,t]$, avec C = c_1 c_2 ……….c_t

  Si e_i=1 alors m_i= 0 ; sinon m_i=1

Le chiffrement de Goldwasser-Micalli est un chiffrement XOR Homomorphe, qui n'a pas d'application concrète actuellement.

Supposons qu'on a c_1 et c_2 deux chiffrés de m_1 et m_2, on aura donc:

$$Dec_{sk}(Enc\ (p_k,m_1).\ Enc\ (p_k,m_2))=Dec_{sk}(Enc_{pk}(z^{m1+m2}.r_1^{\ 2}\ r_2^{\ 2}\ \bmod n)$$

$$= m_1 \oplus m_2 \text{.......................................}(15)$$

I-8-2 Chiffrement Homomorphe multiplicatif :

Par analogie avec ce qui précède, un système basé sur le chiffrement homomorphe multiplicatif permet de n'effectuer que des produits sur les clairs, sans disposer de la clé secrète.

Définition :

Un chiffrement homomorphe est multiplicatif si :

Enc (x $\otimes$ y) = Enc (x) $\otimes$ Enc (y)…………………………………………(16)

$\prod_{i=1}^{n}$ Enc (m_i) = Enc ($\prod_{i=1}^{n}$ m_i)……………………………………………(17)

En d'autres termes, soient :

Enc_p une fonction de chiffrement à clé publique p.

Dec_s une fonction de déchiffrement à clé secrète s.

Alors :

$Dec_s(Enc_p(m) * Enc_p(n))$ = m * n…………………………(18)

Parmi les algorithmes Homomorphes permettant ce type d'opération, nous citons RSA et El Gamal**. [2]**

I-8-2-1 Le chiffrement Homomorphe de RSA : [5]

Le premier système à clé publique à être proposé fut celui de Ronald Rivest, Adi Shamir et Leonard Adleman connu sous le nom RSA. Cet algorithme a été décrit en 1978. Parmi tous les systèmes cryptographiques asymétriques à l'heure actuelle, RSA est considéré comme un des plus solides, si ce n'est le plus solide. Il a résisté à des années de cryptanalyse intensive et il est encore jugé assez robuste pour protéger les échanges bancaires et autres données critiques. Ce niveau de sécurité réside dans la difficulté de factoriser des grands nombres. Retrouver le texte en clair à partir d'une clé et du texte chiffré est supposé équivalent à la factorisation du produit des deux nombres premiers.

Les étapes de la génération des clés de chiffrement et déchiffrement de l'algorithme de RSA sont les suivantes :

Génération des clés :

- On Choisit deux nombres premiers p et q et on calcule le produit n=pq ;

- On Choisit ensuite une clé de chiffrement aléatoire e, tel que e et

(p-1)(q-1) soient premiers entre eux ;

- Finalement, on calcule la clé de déchiffrement de telle manière que :

$$d = e^{-1} \bmod ((p \neq 1)\ (q \neq 1)) \quad \text{.....................................(19)}$$

La clé publique est donc formée des deux nombres e et n, la clé privée est le nombre d.

Ci-dessous l'algorithme de RSA en détail :

Crypto système de RSA :

- **Génération des clés :**
  - Choisir p et q
  - Calculer n=pq ;$\emptyset(n) = (p-1)(q-1)$
  - Déterminer d tel que : $e.d \equiv 1 \bmod \emptyset(n)$
    Clé publique : pk = (e, n)
    Clé privée : sk =d

- **Chiffrement : Enc (m, pk)**
  - Calculer $c=m^e \bmod n$

- **Déchiffrement : Dec (c, sk)**
  - Calculer $m= c^d \bmod n$

Malgré sa robustesse, ce système s'avère vulnérable à l'attaque de l'homme du milieu, c'est à dire par interception et remplacement de la clé publique, l'attaquant récupère la clé publique d'un interlocuteur et fournit au second sa propre clé publique à la place.

Supposons qu'on a c1 et c2 deux textes chiffrées avec l'algorithme de RSA et m1 et m2 les textes clairs correspondants tel que :

$$c_1 = m^e{}_1 \bmod n \text{……(20)}$$

$$c_2 = m^e{}_2 \bmod n \text{……..(21)}$$

$$c_1.c_2 = m^e{}_1 m^e{}_2 \bmod n$$

$$= (m_1 m_2)^e \bmod n \text{……(22)}$$

En déchiffrant le produit des chiffrés, on obtient le produit des clairs :

$$Dec_s(c_1c_2) = ((m_1m_2)^e)^d \bmod n$$

$$= m_1m_2 \text{……(23)}$$

Donc, le chiffrement de RSA réalise la propriété du chiffrement Homomorphe multiplicatif.

I-8-2-2 Le chiffrement Homomorphe d'El Gamal : [6]

Le crypto système d'El Gamal est une méthode de cryptographie à clé publique inventée par Taher ElGamal en 1985. Sa sécurité repose sur la difficulté de calculer le logarithme discret.

La génération des clés de chiffrement et de déchiffrement pour le crypto système d'El Gamal se fait comme suit :

Génération des clés :

- On choisit un nombre premier p et deux nombres aléatoires g et x, tel que g et x soient inférieurs à p ;

- On calcule $y = g^x \bmod p$

La clé publique est donc formée d'y, g et p, la clé privée est x.

L'algorithme d'El Gamal en détail :

Crypto système d'el Gamal :

- **Génération des clés :**
  - Choisir p premier, g et x aléatoires tel que (g, x, < p)
  - Calculer $y=g^x \bmod p$
    Clé publique : pk= (g, p)
    Clé privée : sk =x
- **Chiffrement : Enc (m, pk, k)**
  - Choisir un entier r
  - Calculer $k= g^r \bmod p$
  - Le message chiffré est c'=(k,c)
- **Déchiffrement : Dec(c, sk, k)**
  - Calculer $m= cK^{-x}$

Notons que ce calcul dépend du choix de $K = g^r \bmod p$ (Voir l'algorithme ci-dessus), et donc pour un message clair donné, il y a plusieurs messages chiffrés correspondants, aussi pour chaque message chiffré il faut envoyer un second élément nécessaire au déchiffrement (K).

Supposons qu'on a c_1 et c_2 deux textes chiffrées avec l'algorithme d'El Gamal et m_1 et m_2 les textes clairs correspondants tel que :

$c_1 = (m_1y^r, g^r)$, $c_2 = (m_2y^r, g^r)$

$$c_1c_2 = (m_1m_2.y^{2r}, g^{2r}) \text{.....(24)}$$

Le déchiffrement du produit se fait de la sorte :

$$\text{Decs}(c_1c_2) = m_1m_2.y^{2r} / g^{-2rx}$$

$$= m_1m_2.g^{2xr} / g^{-2xr}$$

$$= m_1m_2 \text{.....(25)}$$

On constate que le chiffrement d'El Gamal réalise aussi la propriété du chiffrement

Homomorphe multiplicatif.

I-8-3 Chiffrement complètement Homomorphe :

Contrairement au chiffrement partiellement homomorphe, avec le chiffrement complètement homomorphe nous pouvons réaliser tout type de calcul sur les données chiffrées stockées dans le Cloud sans les déchiffrer. L'application de ce chiffrement complètement Homomorphe constitue une brique importante dans la sécurité du Cloud.

Plus généralement, on pourrait sous-traiter des calculs sur des données confidentielles à des serveurs situés dans Cloud tout en gardant la clé secrète qui permet de déchiffrer le résultat du calcul.

En 2014, le chiffrement homomorphe devient très prometteur : la commission européenne appelle dans son dernier appel à projet ICT à utiliser le chiffrement homomorphe dans des applications à l'horizon 2020. Le projet HEAT a réuni avec succès les chercheurs de pointe sur ce sujet en Europe (des universités de Leuven, Bristol et du Luxembourg) ainsi que des partenaires industriels spécialisés en cryptographie avancée (Crypto Experts, NXP et Thales) intéressés par le chiffrement homomorphe **[2].**

Définition :

Un système de chiffrement complètement homomorphe est un crypto système permettant de faire des calculs sur les données chiffrées sans les déchiffrer.

Formellement, si c_1(respectivement c_2) est un chiffrée de m_1 (respectivement m_2) il existe deux opérations $\boxdot$ et $\odot$ tel que :

$$Dec(c_1 \boxdot c_2) = Dec(c_1) \odot Dec(c_2) = m_1 \odot m_2 \quad \text{.....(26)}$$

Le chiffrement complètement Homomorphe a été initié par Craig Gentry, ensuite DGHV une nouvelle version de son algorithme appliquée sur les entiers a vu le jour en 2010.

I-8-3-1 Chiffrement de Craig Gentry : [7]

La première construction d'un système complètement homomorphe a été décrite par Gentry en 2009, où il utilise des idéaux d'anneaux de polynômes. La sécurité de ce schéma repose sur les réseaux idéaux.

Pour chiffrer un message, l'idée est d'ajouter du bruit, c'est-à-dire des petites erreurs. La clé secrète permet de supprimer ce bruit, à condition qu'il ne soit pas trop gros. Les opérations homomorphes qui sont effectuées impactent également ce bruit, les bruits vont grossir. On ne pourra déchiffrer le message que si les bruits initiaux sont choisis très petits. Pour dépasser cette limitation sur le nombre d'opérations, et lorsque le bruit devient trop

important, Gentry applique la méthode de "bootstraping" ou d'amorçage. Si le déchiffrement était suffisamment efficace, on pouvait alors changer de clé publique pour réduire le bruit. On commence par utiliser une première clé, puis, quand le bruit devient trop important, on utilise une seconde clé pour rechiffrer le même message.

Le bruit ou (l'erreur) e dans un idéal **I** d'un anneau **R** est définit par : e = kI € I c R. Le message est alors chiffré en ajoutant ce bruit au message.

$$Enc(m) = c = m + kI \text{.....(27)}$$

La procédure de déchiffrement consiste à retirer l'erreur. Les propriétés homomorphes du système sont réalisées, pour :

$$c1 = m1 + k1I \quad et \quad c2 = m2 + k2I$$

On a:

$$c1 + c2 = m1 + m2 + (k1 + K2)\ I$$

ET

$$c1 * c2 = m1 * m2 + (m1k2 + m2k1 + k1k2)I \text{.....(28)}$$

On peut déjà remarquer que le bruit est beaucoup plus affecté par une multiplication que par une addition. Approximativement, une addition double le bruit alors qu'une multiplication l'élève au carré. Si un trop grand nombre d'opérations est effectué, le bruit devient trop grand et la procédure de déchiffrement retourne un message erroné. Cependant, en évaluant régulièrement la procédure de déchiffrement de manière homomorphe, on peut éviter que cela arrive, et c'est exactement ce que fait le bootstraping : Etant donné un chiffré **c** de **m**, cette procédure retourne un chiffré **c'** de **m** où le bruit **k'** contenu dans **c'** est plus petit que le bruit **k** contenu dans **c** : IK'I<IKI.

Cependant, pour pouvoir évaluer la fonction de déchiffrement de façon homomorphe, il est nécessaire que celle-ci soit suffisamment simple, ce qui n'est pas le cas initialement. Pour faire face à ce problème, Gentry réduit la complexité du circuit de déchiffrement en publiant un ensemble de vecteurs dont la somme d'une partie d'entre eux est égale à la clé secrète. Ce problème est connu sous le nom de "[Sparse] Subset Sum Problem", et est prouvé NP-complet.

L'idée de Gentry est de partir d'un schéma dit "somewhat homomorphic encryption scheme" qui peut évaluer des additions et des multiplications tant que le bruit n'est pas trop grand, et de lui appliquer la procédure de bootstrap. Le schéma initial est basé sur le "Ideal Coset Problem". Cependant, pour appliquer la procédure de bootstrap en réduisant la complexité du circuit de déchiffrement il se base sur le "Sparse Subset Sum Problem".

I-8-4 Chiffrement partiellement Homomorphe : [2]

Définition :

On dira d'un crypto système qu'il est partiellement homomorphe lorsque son espace de fonctions évaluables est une restriction de l'espace des fonctions calculables.

Par exemple, le crypto système à clé publique RSA est partiellement homomorphe vis-à-vis de la multiplication. En effet, si (x,y) est un couple d'entiers, on a alors l'égalité suivante :

$\varepsilon(x).\varepsilon(y)=x^e.y^e \bmod N$……(29)

$=(x.y)^e \bmod n$

$=\varepsilon(x,y)$

Nous avons déjà détaillé dans les sections précédentes les algorithmes homomorphes : RSA, El Gamal, Paillier, Goldwasser-Micalli, Gentry et DGHV, dans ce qui suit, nous allons détailler d'autres algorithmes partiellement homomorphes : Benaloh, Okamot-Uchiyama, Sander-Young-Yung et Schmidt Samoa-Takagi**.**

I-8-4-1 Chiffrement de Benaloh :

L'algorithme de Benaloh est détaillé ci-dessous :

Algorithme Benaloh :

Génération des clés : r la taille des blocs, p et q deux grand nombres premiers tel que :

$r \mid p-1$, $pgcd(r,\frac{p-1}{r}) = 1$ et $pgcd(r,q-1)=1$

- Calculer : $N = pq$
- Choisir : $y \in Z^*_N$ au hasard tel que : $y^{\frac{(p-1)(q-1)}{r}} \bmod N \neq 1$

La clé publique : (y,r,N)

La clé privée : (p,q)

Chiffrement : pour m $\in Z_r$

- Choisir u $\in Z^*_N$ au hasard
- Calculer $c = y^m u^r \bmod N$

Déchiffrement :

On doit faire une recherche exhaustive pour trouver quel $i \in \{0, \ldots, p-1\}$

Vérifie : $(y^{-1} c \bmod N)^{\frac{(p-1)(q-1)}{r}} \bmod N = 1$

Le chiffrement de Benaloh est homomorphe pour l'addition :

$$Dec_{sk}((c_1 * c_2) \bmod N) = (m_1 + m_2) \bmod r \quad \ldots\ldots(30)$$

I-8-4-2 Chiffrement d'Okamot-Uchiyama :

Ci-dessous l'algorithme détaillé d'Okamot-Uchiyama :

Okamo-Uchiyama :

Génération des clés :

- Choisir p et q deux nombres premiers de k bits
- Calculer $N = p^2 q$
- Choisir $g \in Z^*_N$ au hasard tel que :
  $g^p \bmod p^2 \neq 1$

La clé publique :(N, g, h,k)

La clé privée : (p, q)

Chiffrement : pour m $\in \{\mathbf{1}, \ldots, \mathbf{2^{k-1}} - \mathbf{1}\}$

- Choisir : $r \in Z^*_N$ au hasard
- Calculer : $c = g^m h^r \bmod N$

Déchiffrement :

$$m=\frac{c^{p-1} \bmod p^2}{g^{q-1} \bmod p^2} \bmod p \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(31)$$

Le chiffrement d'Okamot-Uchiyama est homomorphe pour l'addition :

$$Dec_{sk}((c1 \otimes c2) \bmod N) = (m1 + m2) \bmod N \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(32)$$

I-8-4-3 Chiffrement de Sander-Young-Yung :

L'algorithme de Sander-Young-Yung se présente ainsi :

Sander-young-yung

- **Génération des clés :** se base sur le même principe de l'algorithme de Goldwasser-Micalli
  - Choisir p et q deux grands nombres premiers
  - Calculer N= pq
  - Choisir aléatoirement un non-résidu quadratique y tel que : $\frac{y}{n}=+1$

    La clé publique :(N,y)

    La clé privée : (p, q)

- **Chiffrement : pour chiffrer un message m** $\in \{\mathbf{0}, \mathbf{1}\}$
  - Choisir un entier l>= 1
    - Si m=0 ; choisir un vecteur aléatoire m' $\in (Z_2)^l \neq 0$
    - Si m=1 ; $m^l=0^l$
  - Utiliser la fonction de chiffrement de Goldwasser-Micalli pour chiffrer les m_i pour tout 1<= i>= l

    C= ($Enc_{GM}(b'),\ldots,Enc_{GM}(b_l')$)
- **Déchiffrement :**

  ∀ 1<= i>= l
  - d_i=1 si c_i est un carré modulo N, 0 sinon
  - Si $d= 0^l$ alors m=1 sinon m=0

Le chiffrement de Sander-Young-Yung est homomorphe pour la multiplication :

$$Decsk((c1 \ ¢ \ c2) \bmod N) = (m1 \lozenge m2) \bmod 2 \quad \ldots\ldots\ldots\ldots\ldots\ldots(33)$$

Où ¢ correspond à une multiplication coordonnée par coordonnée.

I-8-4-3 Chiffrement de Schmidt Samoa-Takagi :

L'algorithme de Schmidt Samoa-Takagi en détail :

Schmidt Samoa- Takagi

- **Génération des clés :**
  - Choisir p et q tel que $p \dagger q-1$ et $q \dagger p-1$
  - Choisir l tel que $2^l < pq > 2^{l+1}$
  - Calculer $d=N^{-1} \bmod (p-1)(q-1)$

  La clé publique : (N, l)

  La clé privée :(p, q,d)

- **Chiffrement : pour chiffrer un message m** $\in Z_2^l$
  - Choisir un entier $r \in Z_N^*$ au hasard
  - Calculer $c= r^N(1+mN) \bmod N^2$

- **Déchiffrement :**
  - Calculer $r= c^d \bmod pq$
  - Calculer $\frac{(r^{-N} \bmod N^2)-1}{N} \bmod pq$

Le chiffrement de Schmidt Samoa-Takagi est homomorphe pour l'addition :

$$\mathbf{Dec_{sk}((c_1 * c_2) \bmod N^2) = (m_1 + m_2) \bmod pq} \quad \mathbf{(34)}$$

Aussi:

$$\mathbf{Dec_{sk}(c^k_1 \bmod N^2) = km_1 \bmod pq} \quad \mathbf{(35)}$$

I-9 Conclusion:

Les systèmes de chiffrement homomorphes sont d'une grande importance. Ils offrent la possibilité de traiter des données en tout anonymat en respectant ainsi la vie privée à l'égard des utilisateurs propriétaires de ces données.

De nombreux crypto systèmes homomorphes ont des applications uniquement en théorie mais en pratique dès qu'on veut effectuer un calcul surtout dans le cas de la multiplication, la taille du produit des chiffrés explosent, surtout dans le cas du chiffrement complètement homomorphe qui demande encore des optimisations au niveau des paramètres, chose qui rend son application impossible jusqu'au jour d'aujourd'hui.

Dans ce chapitre nous avons détaillé les algorithmes de chiffrement simplement et complètement homomorphes et dans la suite on va expliquer le le cloud computing véhiculaire dans le chapitre 2.

Chapitre II :

CLOUD COMPUTING Véhiculaire

1- Introduction :

La mise en réseau véhiculaire est devenue un domaine de recherche important en raison de ses caractéristiques spécifiques et des applications telles que la normalisation, la gestion efficace du trafic, la sécurité routière. Les véhicules sont attendus pour transporter les systèmes de communication relativement plus encore, sur les installations informatiques de société, le stockage et l'augmentation de puissance de détection. Par conséquent, plusieurs technologies ont été déployées pour maintenir et promouvoir les systèmes de transport intelligents (STI). Récemment, un certain nombre de solutions ont été proposées pour relever les défis et les enjeux des réseaux de véhicules. Cloud Computing Véhiculaire (CCV) est l'une des solutions. CCV est une nouvelle technologie hybride qui a un impact remarquable sur la gestion du trafic et la sécurité routière en utilisant instantanément les ressources de véhicules, telles que l'informatique, le stockage et Internet pour la prise de décision. Ce chapitre présente les caractéristiques et les modèle de Cloud computing ensuite une section détailler sur l'architecture et les protocoles de routages des réseaux sans fil véhiculaires et une partie explicative sur le Cloud computing véhiculaire **[8].**

Partie I : Le Cloud computing

1. Définition : [8]

L'Institut national des normes et de la technologie (NIST), donne une définition formelle de CC: « Le Cloud Computing est un modèle pour permettre l' accès au réseau pratique, à la demande à un pool partagé de ressources informatiques configurables (par exemple, les réseaux, les serveurs, le stockage, applications et services) qui peuvent être rapidement activé et désactivé avec une interaction minimum d' effort de gestion ou fournisseur de services ».

Le Cloud Forum informatique mobile définit MCC comme suit : « Mobile Computing-Cloud à sa plus simple fait référence à une infrastructure où à la fois le stockage des données et le traitement des données se produisent en dehors de l'appareil mobile. Applications cloud mobiles se déplacent la puissance de calcul et de stockage de données loin de téléphones mobiles et dans le nuage, ce qui porte les applications et l'informatique mobile aux utilisateurs non seulement smartphone, mais une gamme beaucoup plus large d'abonnés mobiles ».

2- Les caractéristiques essentielles du Cloud Computing sont : [9]

2-1- Libre-service à la demande :

Un client peut réserver unilatéralement des capacités informatiques, comme du temps serveur et du stockage, en fonction de ces besoins et de manière automatique, sans nécessiter une interaction humaine avec chaque fournisseur de services.

2-2- Accès ubiquitaire au réseau :

Les capacités sont disponibles au travers du réseau et les accès à ces capacités se font via des mécanismes standards qui promeuvent une utilisation depuis des plateformes clientes légères ou lourdes hétérogènes (par exemple téléphones mobiles, ordinateurs portables et assistants personnels).

2-3- Mise en commun des ressources :

Les ressources informatiques des fournisseurs sont mises en commun de manière à servir plusieurs clients au travers d'un modèle multi locataire, avec les différentes ressources virtuelles et physiques allouées dynamiquement et réaffectées en fonction de la demande du client. Il existe une notion d'indépendance de lieu car le client n'a généralement aucun contrôle ni connaissance de l'emplacement exact des ressources fournies, bien qu'il puisse être en mesure de préciser un lieu à un niveau d'abstraction plus élevé (par exemple un pays, un Etat ou un datacenter). Un espace de stockage, du temps de calcul, de la mémoire, de la bande passante réseau et des machines virtuelles sont autant d'exemples de ressources.

2-4- Elasticité rapide :

Les capacités peuvent être mises à la disposition de manière rapide et élastique, voire automatiquement, pour répondre à une augmentation d'échelle et libérées rapidement en cas de réduction d'échelle. Pour le consommateur, les capacités disponibles apparaissent souvent illimitées et peuvent être achetées en toute quantité à tout moment.

2-5- Service mesuré :

Les systèmes de Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant des possibilités de mesures à un niveau d'abstraction adapté au type de service (par exemple espace de stockage, temps de calcul, bande passante et comptes d'utilisateurs actifs). L'usage des ressources peut être surveillé, contrôlé et indiqué de manière à offrir une certaine transparence au fournisseur et au client du service. Les fournisseurs de cloud computing qui ne parviennent pas à assurer la transparence et à fournir une ou plusieurs des caractéristiques essentielles énumérées ci-dessus, ce qui les distingue des hébergeurs et des fournisseurs de service traditionnels, (par exemple : un rapport détaillé de la consommation par service de vos services consommés, la disponibilité des ressources en cas de besoin) ne sont pas de vrais fournisseurs de cloud computing.

Ces caractéristiques doivent être respectées dans tous les modèles de services offerts par ces fournisseurs.

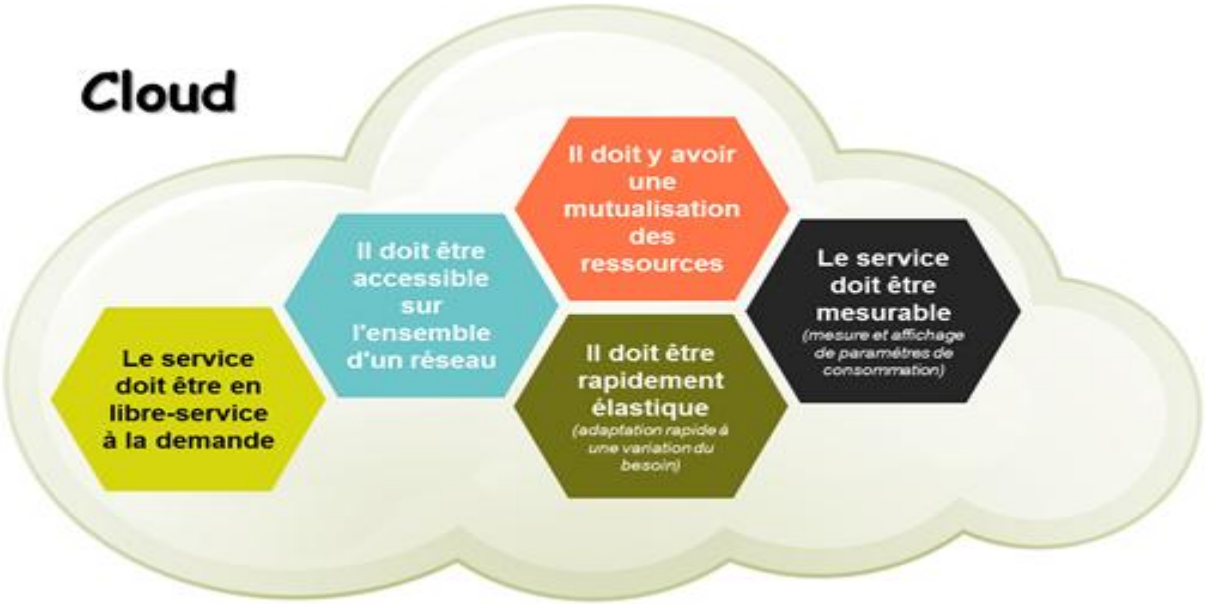


Figure II-1 : les caractéristiques du cloud computing

3- les Trois modèles de services du Cloud Computing : [10]

L'adoption rapide du modèle du Cloud Computing a commencé en 2009. Les entreprises dans tous les pays ont détecté cette opportunité économique chose qui les a obligée à accélérer les manœuvres, la **Figure II.2** montre l'optimisation des capacités IT par rapport à la charge réelle sans et avec le modèle Cloud, par ailleurs la réduction des coûts (les investissements initiaux), l'agilité, la collaboration, l'innovation et la transformation IT en quelques minutes au lieu de quelques jours voire quelques mois sont les principaux leviers de ce nouveau concept.

.

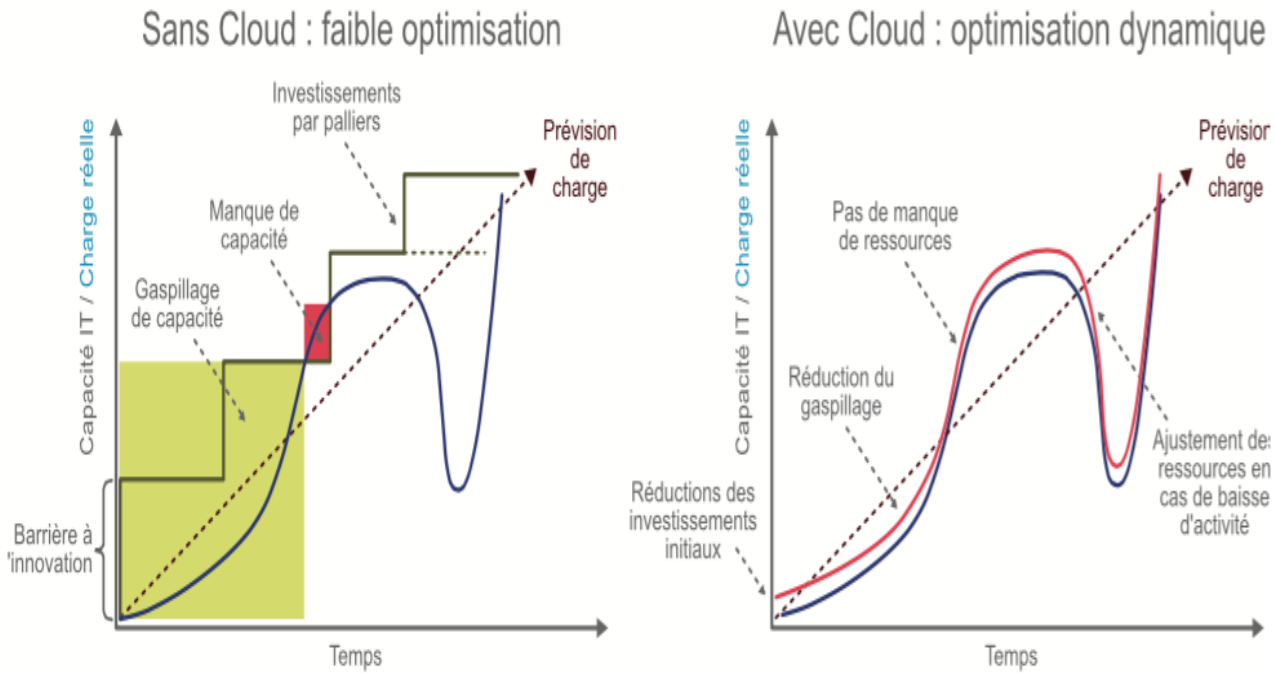


Figure II.2 : Optimisation des ressources avant et après l'adoption du Cloud

Les fournisseurs du Cloud proposent leurs services sous différents modèles que NIST a énumérés selon les types de fonctionnalités et les besoins des entreprises.

3-1- Software as a Service (SaaS)

Dans ce modèle, le client est en capacité d'utiliser les applications du fournisseur qui s'exécutent dans une infrastructure de Cloud, les applications sont accessibles à partir de différents périphériques clients au travers d'une interface légère (navigateur web par exemple).

Le client ne gère ni ne contrôle l'infrastructure de Cloud sous-jacente, comme : le réseau, les serveurs, les systèmes d'exploitation, le stockage ou les possibilités de déployer ses propres applications, mais il pourra éventuellement configurer certains paramètres de l'application destinés à l'utilisateur.

C'est la fourniture clé en main d'applications prêtes à l'emploi, avec une prise en charge du paramétrage du service concerné et une facturation en abonnement sur la base d'une tarification à l'usage, historiquement le SaaS est un peu la première brique Cloud apparue.

Voici quelques exemples de domaines d'utilisation du SaaS :

- Messagerie électronique

- CRM (Customer Relationship Management)

- GED (Gestion électronique de documents)

- Collaboration en ligne

- Logiciels de gestion de paie et RH

- Sauvegardes en ligne

Les principales offres SaaS proposées sont**:**

- Google offre Google Apps (messagerie et bureautique)

- Sales Force CRM (Customer Relationship Management)

- Microsoft offre Office 365 (messagerie, outils collaboratifs, bureautique)

3-2- Plateforme as a Service (PaaS) :

Dans ce modèle, le client est en capacité de déployer dans l'infrastructure du Cloud des applications qu'il a créées ou qu'il a achetées et qui ont été développées à l'aide des langages de programmation et des outils pris en charge par le fournisseur. Le client ne gère ni ne contrôle l'infrastructure de Cloud sous-jacente, comme le réseau, les serveurs, les systèmes d'exploitation ou le stockage, mais dispose d'un contrôle sur les applications déployées et éventuellement sur l'environnement qui les héberge.

Le PaaS propose à l'utilisateur, en plus d'un service d'utilisation de logiciel à distance, d'avoir accès à une véritable plateforme de développement, équipée d'un langage de programmation, d'outils de développements, de modules. L'utilisateur bénéficie donc d'un environnement de développement managé, hébergé, maintenu par un fournisseur Cloud, basé sur une infrastructure externe à son entreprise. Il aura donc la possibilité de développer des outils uniques pour son activité à l'aide d'une interconnexion collective de plusieurs intervenants internes ou externes.

Les principales offres PaaS proposées sont :

- Microsoft avec Windows AZURE

- Google avec Google App Engine

- Orange Business Services.

3-3- Infrastructure as a Service (IaaS) :

Dans ce modèle, le client est en capacité d'approvisionner des ressources de calcul, de stockage, de réseau et d'autres, où il est en mesure de déployer et d'exécuter des logiciels quelconques, comme des systèmes d'exploitation et des applications. Le client ne gère ni ne contrôle l'infrastructure de Cloud sous-jacente, mais dispose d'un contrôle sur les systèmes d'exploitation, le stockage, les applications déployées et éventuellement sur l'ensemble des composants réseau sélectionnés (par exemple le pare-feu de l'hôte). La définition du service comprend des offres telles que l'espace serveur, des connexions réseau, de la bande passante, des adresses IP et des load balancers. Physiquement, les ressources hardware proviennent d'une multitude de serveurs et de réseaux généralement distribués à travers de nombreux Datacentres, que le fournisseur de services Cloud a la responsabilité d'entretenir. Parallèlement, l'accès aux composants virtualisés est donné à l'entreprise cliente afin que celle-ci puisse construire ses propres plateformes IT.

Les principales offres IaaS proposées sont :

- Amazon Web Services (AWS) avec Elastic Compute Cloud (EC2)

- Microsoft avec Azure

- Google avec Compute Engine

Le Cloud Computing est une solution qui fournit un espace dans lequel il est possible de placer virtuellement des infrastructures serveur ou réseau, des plateformes de développement ou d'exécution, et tout cela peut être déployé sur différentes typologies du Cloud appelées les modèles de déploiement du Cloud Computing.

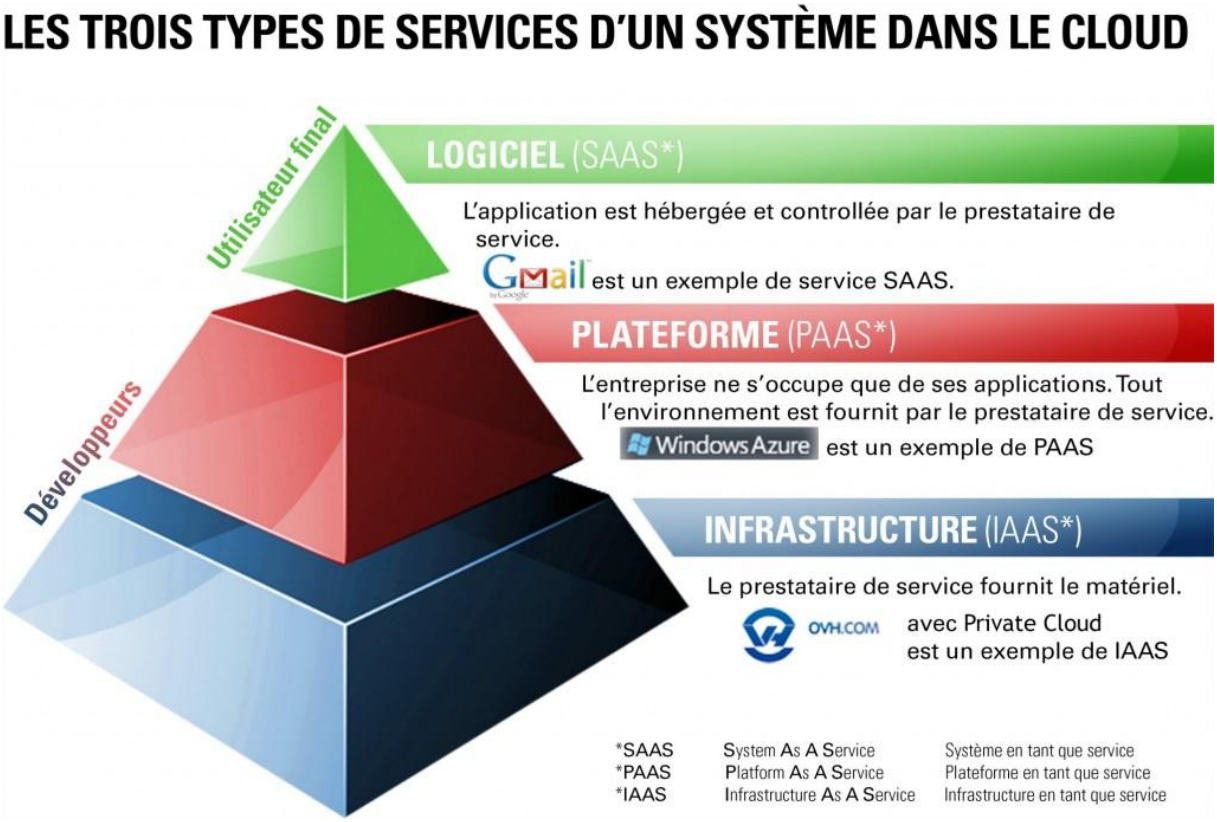


Figure II-3 : les modèles du cloud computing

4- les Quatre modèles de déploiement du Cloud Computing : [11]

Le NIST a défini quatre modèles de déploiement du Cloud Computing qui peuvent faire l'objet de variantes importantes en fonction d'autres facteurs que nous allons détailler dans les sections suivantes (sécurité, sensibilité des données, prix...)

4-1- Le Cloud privé :[2]

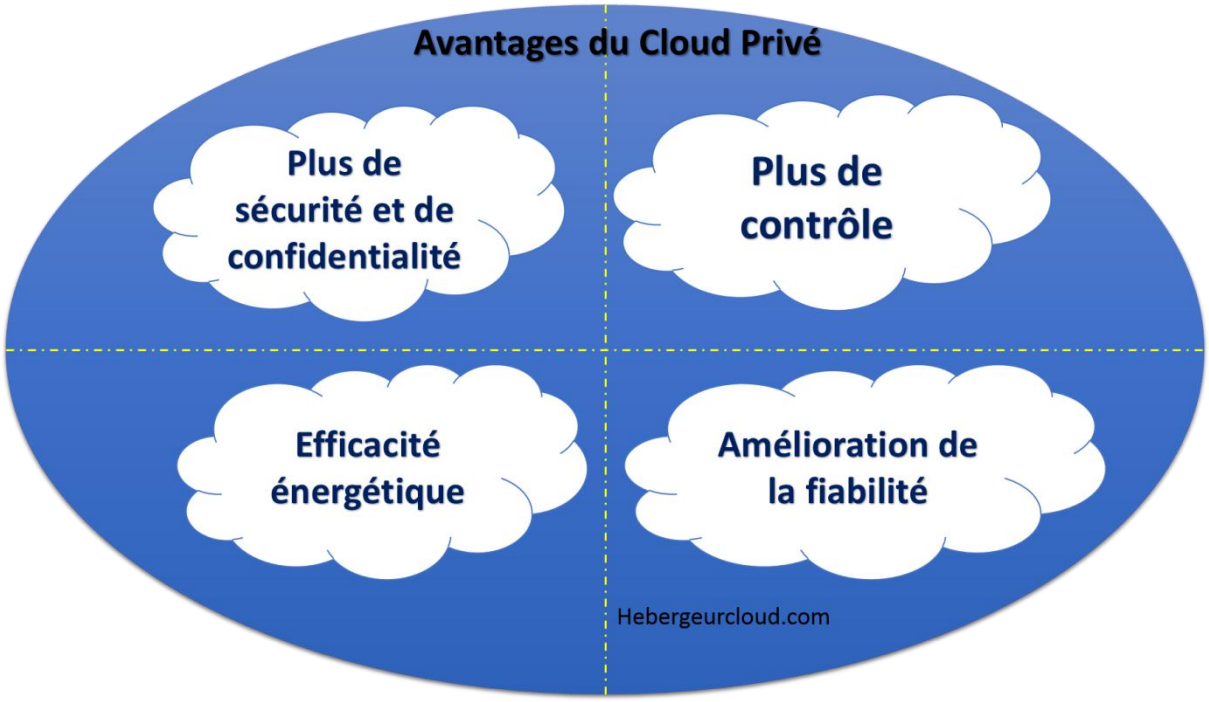
L'infrastructure du Cloud est réservée à une entreprise. Elle peut être gérée par l'entreprise ou par un tiers et peut se trouver dans les locaux de l'entreprise ou ailleurs.

Les deux caractéristiques du Cloud privé sont la délimitation d'un Cloud pour l'utilisation d'une seule organisation ainsi qu'un degré plus élevé de sécurité du réseau.

Les ressources des services du Cloud privé proviennent d'un pool distinct de serveurs physiques, pouvant être hébergés en interne ou hors de l'entreprise, et accessibles par des liaisons louées privées ou des connexions sécurisées via les réseaux publics.

La sécurité supplémentaire que fournit le modèle de Cloud privé est idéale pour tout type d'organisation ou d'entreprise ayant besoin de stocker et de traiter des données privées, ou alors d'exécuter des tâches sensibles. Par exemple, un service de Cloud privé peut typiquement être utilisé par une société financière obligée par la législation en vigueur de stocker des données sensibles internes, mais qui souhaite également avoir recours à certains avantages du Cloud computing dans son infrastructure, comme l'allocation de ressources sur demande.

Les avantages du cloud privé :



4-2- Le Cloud Communautaire :[2]

L'infrastructure du Cloud est partagée par plusieurs entreprises et est destinée à une communauté précise aux préoccupations communes (par exemple une mission, des exigences de sécurité, une stratégie ou des questions de conformité). Elle peut être gérée par des entreprises ou un tiers et peut se trouver dans leurs locaux ou ailleurs.

C'est un modèle dédié à une communauté professionnelle spécifique incluant partenaires, sous-traitants..., pour qu'il puisse travailler de manière collaborative sur un même projet ou il peut s'agir d'un Cloud gouvernemental dédié aux institutions étatiques.

4-3- Le Cloud public :[2]

L'infrastructure du Cloud est rendue disponible au grand public ou à un grand groupe industriel et elle appartient à une entreprise qui vend des services en nuage.

C'est le modèle le plus connu vis-à-vis les utilisateurs Cloud. Les services sont fournis dans un environnement virtualisé, construit en utilisant des ressources physiques partagées et accessibles via un réseau public (Internet). Le principe du Cloud public peut être défini par opposition à celui du Cloud privé, via lequel de nombreux clients accèdent à des services virtuels qui tirent tous leurs ressources du même pool de serveurs à travers des réseaux publics. En revanche, le Cloud public fournit des services à des clients multiples en utilisant la même infrastructure partagée.

Les offres de Software as a Service (SaaS), comme le stockage Cloud et les applications office en ligne, sont peut-être les plus connues, mais les offres disponibles d'Infrastructure as a Service (IaaS) et de Platform as a Service (PaaS), qui incluent l'hébergement web et des environnements de développement basés sur le Cloud, peuvent également correspondre à ce modèle (bien que toutes puissent exister au sein de Clouds privés). Les Clouds publics sont largement utilisés dans les offres adressées aux grand public, moins susceptibles d'avoir besoin de l'infrastructure et de la sécurité des Clouds privés. Toutefois, les entreprises peuvent toujours avoir recours au Cloud public pour rendre leurs opérations plus efficaces, par exemple pour le stockage de contenu non-sensible, la collaboration avec des documents en ligne et la messagerie web.

4-4- Le Cloud Hybride :[2]

L'infrastructure du Cloud est constituée de deux Clouds ou plus (privés, communautaires ou publics) qui restent des entités indépendantes mais sont reliés par une technologie standardisée ou propriétaires afin d'autoriser une portabilité des données et des applications (par exemple le "Cloud bursting" pour la répartition de charge entre les différents Clouds).

Ce modèle permet la cohabitation entre deux ou plusieurs Clouds privés et des Clouds publics pour remplir différentes fonctions au sein d'une même organisation. Si tous les types de services Cloud sont sensés offrir un certain niveau d'efficacité, à des degrés divers, les services Cloud public sont susceptibles d'être plus avantageux au niveau des coûts et plus évolutifs que les Clouds privés. C'est pourquoi une organisation peut maximiser son efficacité en utilisant des services de Cloud public pour ses opérations non-sensibles et s'appuyer en revanche sur un Cloud privé lorsqu'elle en a besoin, faisant en sorte que toutes ses plateformes soient intégrées harmonieusement.

Les modèles du Cloud hybride peuvent être mis en œuvre par de nombreuses façons :

- Les différents fournisseurs de Cloud s'unissent afin de fournir des services intégrés en Cloud privé et public.

- Les fournisseurs de Cloud individuels proposent un pack hybride complet ;

- Les organisations gérant elles-mêmes leur propre Cloud privé souscrivent à un service de Cloud public qu'elles intègrent ensuite dans leur infrastructure.

Il est important de préciser qu'un client peut obtenir un meilleur contrôle sur la sécurité d'un plus grand nombre de ressources lorsqu'il passe de SaaS au PaaS et plus encore de PaaS à IaaS, de même que lorsqu'il passe d'un Cloud public à un Cloud communautaire et plus encorde' un Cloud communautaire à un Cloud privé.

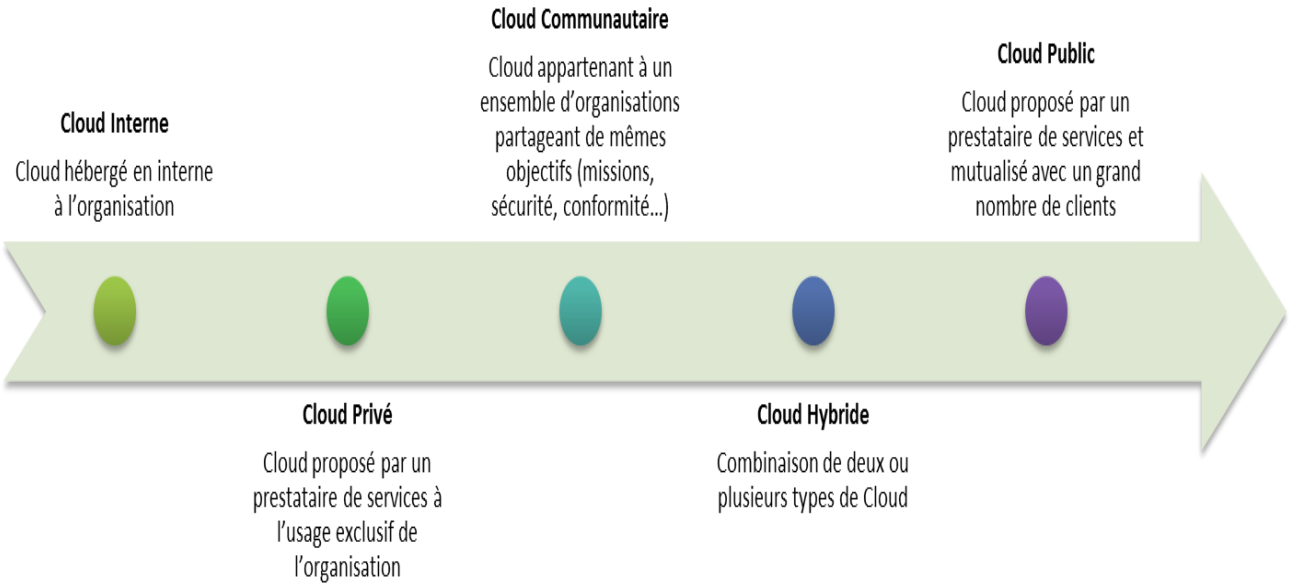


Figure II-4: les modèles du cloud computing

Partie II : Les réseaux VANETs

1 - Définition d'un réseau VANET:[12]

Un réseau VANET est une particularité des réseaux MANET où les nœuds mobiles sont des véhicules (intelligents) équipés de calculateurs, de cartes réseau et de capteurs.

Comme tout autre réseau Ad hoc, les véhicules peuvent communiquer entre eux (pour échanger les informations sur le trafic par exemple) ou avec des stations de base placées tout au long des routes (pour demander des informations ou accéder à internet…).

La **figure II.5** représente la hiérarchie des réseaux sans fil où elle schématise l'inclusion des réseaux véhiculaires Ad Hoc VANET dans les réseaux mobile Ad Hoc MANET, les MANET dans les réseaux Mobiles ainsi que les réseaux mobiles dans les réseaux sans fil.

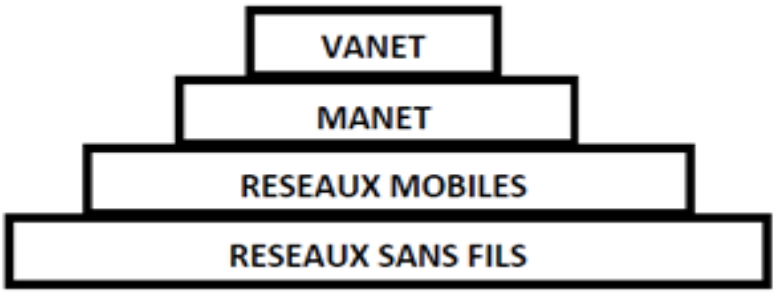


Figure II.5 : Hiérarchie des réseaux sans fils

- **Le nœud du réseau VANET**

Un nœud d'un réseau VANET est un véhicule équipé de terminaux tels que les calculateurs, les interfaces réseaux ainsi que des capteurs capables de collecter les informations et de les traiter. On parle de la notion de « véhicule intelligent ».

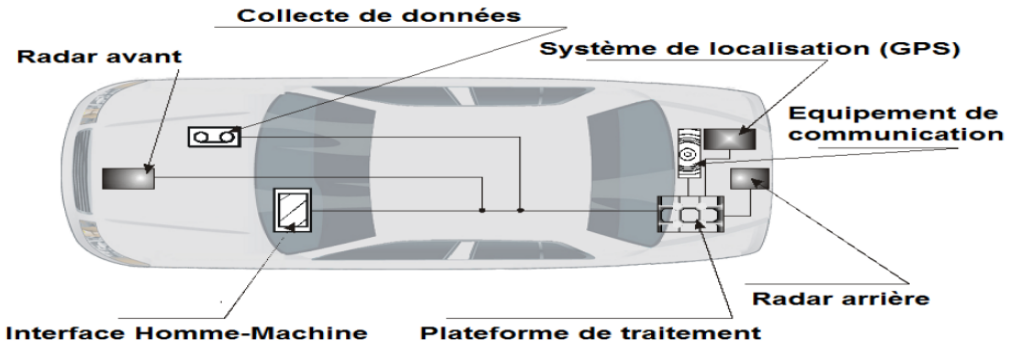


Figure II.6 : Véhicule intelligent

2- Architectures des réseaux sans fil véhiculaires :

Un réseau sans fil véhiculaire est un ensemble d'entités communicantes organisées selon une architecture de communication. Ces entités embarquées peuvent rencontrer différents environnements (urbain, péri-urbain, autoroutier), ayant leurs propres contraintes.

2-1- Architectures de communication :

Dans les réseaux de véhicules, on peut distinguer deux modes de communication, les communications Véhicule-Infrastructure et les communications Véhicule-à-Véhicule.

Les véhicules peuvent utiliser un de ces deux modes ou bien les combiner s'ils ne peuvent pas communiquer directement avec les infrastructures.

1- Mode de communication de Véhicule-Infrastructure :

L'architecture Véhicule-vers-Infrastructure (V2I) est composée de RSU, auxquels les véhicules accèdent pour les applications de sécurité, de gestion et de confort. Les RSU sont administrés par un ou plusieurs organismes publics ou bien par des opérateurs autoroutiers. Un véhicule qui informe le service de voirie au sujet d'un obstacle est un exemple de communication V2I. Dans cet exemple, la communication est unidirectionnelle, du OBU vers le RSU.

Nous parlons d'I2V dans le cas de communication Infrastructure-vers-Véhicule. Un panneau de signalisation équipé d'un RSU qui envoie une information aux véhicules passant à proximité est un exemple de communication I2V. Dans la suite, par V2I, nous englobons toutes les communications Véhicule-Infrastructure, quelle que soit la direction du trafic de données.

L'inconvénient majeur de cette approche est que l'installation des stations le long des routes est une tâche coûteuse et prend beaucoup de temps, sans oublier les coûts relatifs à la maintenance des stations.

2-Mode de communication Véhicule à Véhicule :

L'architecture de communication inter-véhicules (V2V ou IVC pour Inter Véhicule Communication) est composée uniquement d'OBU (véhicules légers, poids lourds, véhicules de secours, etc.). Ils forment alors un réseau mobile sans avoir besoin d'un élément de coordination centralisé. Cette situation est essentielle si certains équipements RSU deviennent indisponibles (en panne ou hors de portée). Dans ce cas, le réseau doit continuer de fonctionner. Les véhicules doivent alors collaborer pour assurer la disponibilité du service. Ce mode de fonctionnement est communément appelé « ad hoc » et est utilisé par les VANETs. L'architecture V2V en mode ad hoc peut aussi être utilisée dans les scénarios de diffusion d'alerte (freinage d'urgence, collision, ralentissement, etc.) ou pour la conduite coopérative. En effet, dans le cadre d'applications de sécurité routière, les réseaux à infrastructure montrent leurs limites, surtout en terme de délai. Prenons l'exemple d'un véhicule en difficulté sur la chaussée qui diffuse un message d'alerte. Il semble plus rapide d'envoyer l'information directement aux autres véhicules plutôt que de la faire transiter par une station de base.

3-Communication Hybride

La combinaison des communications véhicule à véhicule avec les communications de véhicules avec utilisation d'infrastructures, permet d'obtenir une communication hybride très intéressante. En effet, les portées des infrastructures (stations de bases) étant limitées,

l'utilisation des véhicules comme relais permet d'étendre cette distance. Dans un but économique et afin d'éviter la multiplication des stations de bases à chaque coin de rue, l'utilisation des sauts par véhicules intermédiaires prend tout son importance.

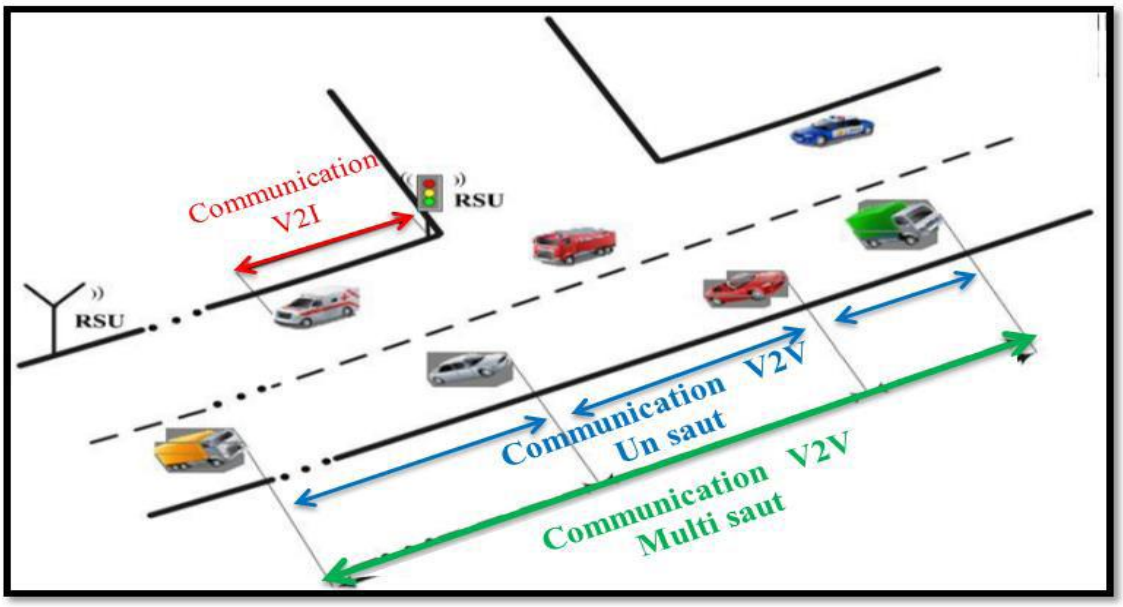


Figure II.7 : Les modes de communication dans les VANETs

3- Routage dans les VANETs : [13]

Le routage joue un rôle très important dans les VANET puisque tous les services supportés, unicast ou multicast, se basent sur des communications multi-saut pour l'acheminement des données. Les transferts de fichiers et les jeux. Les communications multicast sont utilisées dans les applications de sécurité et de gestion de trafic telles que l'avertissement de collision et le platooning. Pour réaliser les échanges, les protocoles de routage utilisent des informations locales, sur le voisinage immédiat, ou globales, concernant tout le réseau, an de déterminer les nouds relais qui participent à l'acheminement des données communications unicast sont généralement utilisées dans les applications de confort.

3-1- Classification des protocoles de routage dans les réseaux VANET :

Les réseaux véhiculaires ont comme caractéristique principale une forte mobilité qui entraîne une topologie très dynamique. Cette caractéristique fait que les protocoles de routage traditionnels des MANETS sont pour la plupart in adaptée aux VANETS. En effet, dans les VANETS, la vitesse peut être beaucoup plus élevée que les MANETS dans certains environnements de communication comme les autoroutes. Dans Différentes solutions pour

le routage dans les réseaux VANET ont été proposées, nous distinguons deux classes de protocoles de routage: les protocoles basés sur la Unicast (topologie) qui sont divisés en protocoles proactifs, réactifs et hybrides et les protocoles basés sur la localisation (géographique) qui utilisent la position physique des nœuds mobiles pour configurer le routage.

3-1-1- Les protocoles de routage basés sur la topologie :

1- Les protocoles réactifs :

Les protocoles réactifs adoptent des algorithmes classiques tels que le routage par vecteur de distance. Les routes sont établies uniquement sur demande et seules les routes en cours d'utilisation sont maintenues. Dans ce cas, un délai supplémentaire est nécessaire au début de chaque session pour la recherche du chemin .Lorsqu'un nœud veut envoyer des paquets, une étape de découverte de route est initiée par la diffusion d'un message de recherche de route. Tout nœud qui reçoit ce message et qui ne dispose pas d'informations à propos de la destination diffuse à son tour le message. Ce mécanisme est appelé mécanisme d'inondation.

a. Le protocole AODV :

Le protocole de routage AODV (Ad hoc On-demand Distance Vector) est un protocole qui crée les routes au besoin et utilise le principe de numéro de séquence afin d'utiliser les routes les plus nouvelles, dites encore les plus fraîches. En plus, il utilise le nombre de sauts comme métrique pour choisir entre plusieurs routes disponibles. Trois types de paquets sont utilisés par AODV : les paquets de requête de route RREQ (Route Request Message), les paquets de réponse de route RREP (Route Reply Message) et les paquets d'erreur de route RERR (Route Error Message). En plus de ces paquets, AODV invoque des paquets de contrôle HELLO qui permettent de vérifier la connectivité des routes. AODV repose sur deux mécanismes : découverte de route et maintenance de route. La découverte de route permet de trouver une route pour atteindre une destination et la maintenance de route permet de détecter et signaler les coupures de routes provoquées éventuellement par la mobilité des nœuds.

b. Le protocole DSR

Le protocole de routage DSR (Dynamic Source Routing) est un protocole qui crée les routes à la demande comme le protocole AODV. Il utilise la technique "routage à la source" dans laquelle la source inclut dans l'entête du paquet la route complète par laquelle un paquet doit passer pour atteindre sa destination. Les nœuds intermédiaires entre la source et la destination n'ont pas besoin de maintenir à jour les informations sur la route traversée puisque la route complète est insérée dans l'entête du paquet. DSR est composé de deux mécanismes : la découverte de route et la maintenance de route. Le premier permet de chercher les routes nécessaires à la demande, tandis que le second permet de s'assurer de la maintenance des routes tout au long de leur utilisation.

2- Les protocoles proactifs

Les protocoles proactifs, chaque nœud garde une image de la topologie de tout le réseau. Cette image est mise à jour, périodiquement ou à chaque modification topologique, par un échange de messages de contrôle. Les routes sont déterminées sur la base de cette image.

a. Le protocole OLSR

Le protocole de routage OLSR (Optimized Link State Routing) est un protocole de routage proactif développé dans le cadre du projet Hypercom de l'Institut National de la Recherche en Informatique et Automatique (INRIA) de France et proposé en tant que RFC (Request For Comment) expérimentale à l'IETF (Internet Engineering Task Force). Il est considéré comme une optimisation du protocole à état des liens filaires pour les réseaux mobiles Ad Hoc. Il a pour objectif de fournir des routes de plus court chemin vers une destination en termes de nombre de sauts en utilisant l'algorithme de Dijkstra. Son innovation réside dans sa façon d'économiser les ressources radio lors des diffusions, ceci est réalisé grâce à l'utilisation de la technique des relais multipoints (MPR : Multi-Point Relaying), donc le principe est que chaque nœud construit un sous ensemble appelé MPR, qui permet d'atteindre tous ses voisins à deux sauts, les nœuds de cet ensemble servent à acheminer et retransmettre les messages qu'ils reçoivent. Les voisins d'un nœud qui ne parmi ses voisins sont pas MPRs, lisent et traitent les paquets mais ne les retransmettent pas.

b. Le protocole DSDV

Le protocole de routage DSDV (Destination-Sequenced Distance-Vector) est un protocole de routage de type vecteur de distance. Chaque nœud maintient une table de routage contenant des informations sur les destinations accessibles dans le réseau. Ces informations comprennent le nœud suivant utilisé pour atteindre la destination, le nombre de sauts qui sépare le nœud de la destination et le numéro de séquence estampillé par la destinataire. Ce numéro de séquence permet de distinguer les nouvelles routes des anciennes. Chaque nœud envoie périodiquement à ses voisins la totalité de sa table de routage. D'autres paquets de mise à jour sont aussi envoyés à la suite d'un changement dans la topologie du réseau. Ces paquets n'incluent que les entrées de la table affectées par le changement et ont pour objectif de propager les informations de routage aussi rapidement que possible. Quand un nœud reçoit un paquet de mise à jour, il le compare avec les informations existantes dans sa table de routage. Toute entrée dans la table est mise à jour si l'information reçue est plus récente (ayant un numéro de séquence plus grand), ou si elles ont le même numéro de séquence mais avec une distance plus courte.

Dans le protocole DSDV, une unité mobile doit attendre jusqu'à ce qu'elle reçoive la prochaine mise à jour initiée par la destination afin de mettre à jour l'entrée associée à cette destination dans la table de distance. De ce fait, la réaction de DSDV aux changements de la topologie est considérée lente. D'autre part, ce protocole cause une charge de contrôle importante dans le réseau à cause des paquets de mise à jour envoyés périodiquement ou à la suite des événements.

c. Le protocole GSR

Le protocole GSR (Global State Routing) est un protocole proactif à état de liens où chaque nœud connaît la topologie globale du réseau ce qui lui permet de calculer les routes pour atteindre chaque destination. GSR diffère des protocoles à état de liens dans le fait que les nœuds ne diffusent pas leurs états de liens à tout le réseau, mais ils se limitent à l'envoyer aux voisins uniquement. Ainsi, GSR réduit le trafic des paquets de contrôle. Le problème de GSR est la taille de ses paquets de mise à jour (table de topologie) qui peuvent devenir considérable si le réseau contient un grand nombre de nœuds. En plus, il a une lenteur dans la détection des changements de la topologie.

3- Protocoles hybrides :

a. Le protocole ZRP

Le protocole de routage ZRP (Zone Routing Protocol) est un protocole hybride qui combine les deux approches proactives et réactive. Le protocole ZRP divise le réseau en différentes zones. Pour chaque nœud, il définit une zone de routage exprimée en nombre de sauts maximal σ. Ainsi, la zone de routage d'un nœud inclut tous les nœuds qui sont à une distance au maximum de σ sauts. Les nœuds qui sont exactement à σ sauts sont appelés nœuds périphériques.

À l'intérieur de cette zone, ZRP utilise un protocole proactif et à l'extérieur de cette zone de routage, il fait appel à un protocole réactif.

Le protocole proactif est IARP (IntrAzone Routing Protocol) et celui réactif est IERP (IntErzone Routing Protocol) Chaque nœud doit tout d'abord connaître ses voisins. Pour cela, ZRP utilise soit le protocole de contrôle d'accès au support (MAC) pour connaître les voisins immédiats ou le protocole NDP (NeighbourDiscovery Protocol) pour la transmission et la gestion des échanges de messages HELLO. Par la suite, chaque nœud invoque le protocole IARP pour découvrir les routes vers tous les autres nœuds qui se trouvent dans sa zone de routage. Cependant, le protocole IERP est utilisé à la demande pour chercher les routes entre un nœud et une destination qui se trouvent à l'extérieur de sa zone de routage. Un troisième protocole BRP (Bordercast Resolution Protocol) est inclus avec IERP pour guider la propagation des requêtes de recherche de route dans le réseau. BRP utilise les données de la topologie fournies par le protocole IARP afin de construire sa liste des nœuds de périphérie et la façon de les atteindre.

L'avantage des protocoles hybrides est le fait qu'ils s'adaptent mieux aux réseaux de grandes tailles. Cependant, ce type de protocole cumule les inconvénients des protocoles proactifs et ceux des protocoles réactifs, tels que l'échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné.

3-1-2- Les protocoles de routage basés sur la géographique :

Les protocoles de routage géographiques sont les plus adaptés pour les réseaux ad hoc de véhicules, puisque le mécanisme de routage se base sur les données géographiques des nœuds.

a- Le protocole A-STAR :

Le protocole de routage A-STAR (Anchor-based Street and Traffic Aware Routing) est un protocole de routage basé sur la localisation (position) pour un environnement de communication véhiculaire métropolitain. Il utilise particulièrement les informations sur les itinéraires d'autobus de ville pour identifier une route d'ancre (anchor route) avec une connectivité élevée pour l'acheminement des paquets. A-STAR adopte une approche de routage basée sur l'ancrage (anchor based) qui tient compte des caractéristiques des rues. Un point est associé à chaque rue en fonction de sa capacité (grande ou petite rue qui est desservie par un nombre de bus différent). Les informations de routes fournies par les bus donnent une idée sur la charge du réseau véhiculaire dans chaque rue. Ce qui donne une image de la ville a des moments différents.

b- Le protocole UMB :

Le protocole de routage UMB (Urban Multi hop Broadcast Protocol) C'est un protocole efficace de la norme 802.11, basé sur l'algorithme de diffusion multi saut pour les réseaux inter véhiculaires avec support d'infrastructure, dans le but de réduire les collisions et d'utiliser efficacement la bande passante. Contrairement aux protocoles de diffusion par inondation, UMB confie les opérations d'envoi et de reconnaissance des paquets aux nœuds les plus éloignés sans connaître à priori des informations sur la topologie du réseau.

UMB est décomposé en deux phases : la première appelée diffusion directionnelle, où le véhicule source sélectionne un nœud dans la direction de diffusion pour faire un relayage de données sans aucune information sur la topologie. La deuxième diffusion aux intersections pour disséminer les paquets dans toutes les directions, pour cela UMB utilise des répéteurs installés dans les intersections pour l'envoi des paquets vers tous les segments. On suppose que chaque véhicule est équipé par un récepteur GPS (Global Position System) et une carte routière électronique. Le principal avantage du protocole UMB est la fiabilité de diffusion multi-saut dans les canaux urbains.

c- Le protocole GyTAR :

Le protocole de routage GyTAR (improved Greedy Traffic-Aware Routing protocol) est un protocole de routage géographique basé sur la localisation (position) et adapté aux réseaux véhiculaires capable de trouver des chemins robustes dans un environnement urbain. L'objectif de ce protocole est de router les données de proche en proche en considérant les différents facteurs spécifiques à ce genre d'environnements/réseaux. Ce protocole suppose que chaque véhicule connaît sa position courante et ceci grâce au GPS. De plus un nœud source est sensé connaître la position du destinataire pour pouvoir prendre des décisions de routage, cette information est donnée par un service de localisation tel que GLS (Grid Location Service) et peut déterminer la position des intersections voisines à travers des cartes numériques.

d- Le protocole VADD

Le protocole de routage VADD (*Vehicle-Assisted Data Delivery*) est un protocole de routage qui prend en considération le contexte des réseaux de véhicules et exploite le mouvement prévisible des véhicules pour décider de retransmettre ou non le message. Il utilise particulièrement les informations sur le trafic routier au niveau d'une route pour estimer le délai mis par un paquet pour parcourir un tel segment. Par conséquent, les paquets seront achemines le long d'un chemin ayant le plus faible délai de bout en bout.

e- Le protocole MORA :

Le protocole de routage MORA (*MOvement-based Routing Algorithm*) exploite la position et la direction de mouvement de véhicules pour adapter les décisions de retransmission au contexte des véhicules et faire face ainsi a la forte mobilité des nœuds et au changement assez fréquent de la topologie.

f- Le protocole GPSR :

Le protocole de routage GPSR (Greedy Perimeter Stateless Routing) est donc un protocole de routage basé sur la position, qui contient deux parties. La première correspond à une méthode de choix du prochain noud transmetteur qui aura le rôle de retransmettre les paquets, et cela tout en se basant sur les informations de position des voisins (nouds candidats) et de la destination des paquets. Cette méthode consiste à choisir le candidat qui est à une distance la plus proche à vol d'oiseau de la destination. La deuxième partie de GPSR est en fait une méthode pour contourner les obstacles et les zones géographiques vides, qui ne présentent aucun candidat transmetteur dans le voisinage.

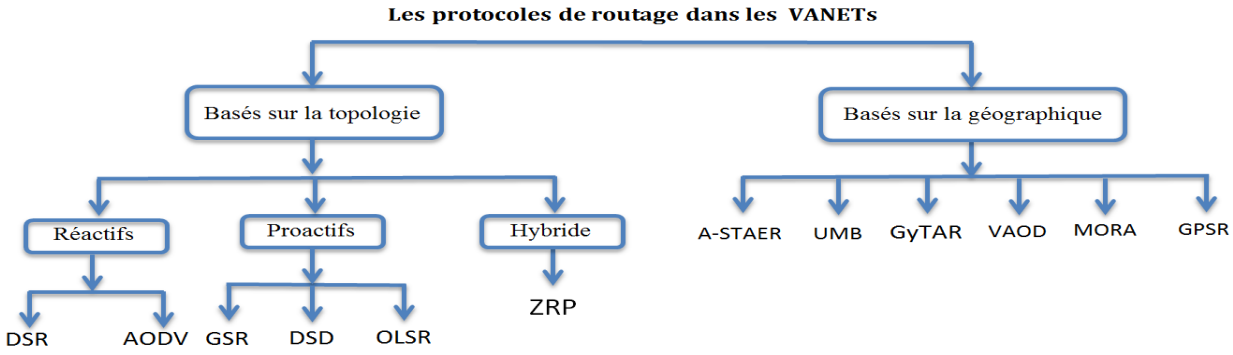


Figure II.8 : Les protocoles de routage dans les réseaux VANETs

4. Application des réseaux VANETs : [14]

Les principales applications des réseaux vanet peuvent être classées en trois Catégories: applications de sécurité routière, applications d'aide à la conduite, et applications de confort.

1- Applications pour la sécurité routière :

La sécurité routière est devenue une priorité dans la plupart des pays développés. Cette priorité est motivée par le nombre croissant d'accidents sur ses routes associé à un parc de véhicules de plus en plus important. Afin d'améliorer la sécurité des déplacements et faire face aux accidents routiers, les IVC offrent la possibilité de prévenir les collisions et les travaux sur les routes, de détecter les obstacles (fixes ou mobiles) et de distribuer les informations météorologiques.

2- Applications pour les systèmes d'aide à la conduite et les véhicules coopératifs :

Pour faciliter la conduite autonome et apporter un support au conducteur dans des situations particulières : aide aux dépassements de véhicules, prévention des sorties de voies en ligne ou en virage, etc. Nous pouvons citer également le cas des compagnies de transports utilisant les IVC dans un but de productivité pour réduire la consommation de carburant;

3- Applications de confort du conducteur et des passagers :

En particulier les services de communication et d'informations des utilisateurs comme l'accès mobile à l'Internet, la messagerie, le chat inter-véhicules, les jeux en réseaux, etc.

Dans la suite de cette partie nous nous limitions à la description de quelques services et exemples d'application des systèmes de communication véhicule à véhicule.

Partie III : cloud computing Véhiculaire : [11]

1- définition :

Eltoweissy et al. (2010a) et Olariu et al. (2011) a introduit le concept d'un nuage véhiculaire (VC) qui tire parti des ressources du conseil dans les voitures participantes. Certains véhicules sont garés pendant de longues heures tandis que d'autres sont coincés dans les embouteillages et se déplacent lentement, en modifiant leur position dans certains réseaux sans fil. Enfin, nos voitures passent beaucoup de temps sur la route et peuvent faire face à des emplacements dynamiquement fluctuant. Dans ce cas, les véhicules aideront consultants locaux à résoudre les incidents de circulation en temps opportun ce qui est impossible avec les centres de gestion du trafic municipal seul en raison du manque de ressources informatiques adéquates. Nous nous attendons à ce que, les véhicules sont capables de résoudre les problèmes dans de nombreuses situations qui peuvent nécessiter une durée indéterminée pour un système centralisé.

En fin de compte, en utilisant des ressources autonomes auto-organisées, les véhicules serviront à la demande en temps réel pour résoudre de grands problèmes graves d'événements imprévus. Les nouveaux nuages de véhicules contribueront à résoudre les défis techniques et de contribuer à des systèmes de transport complexes avec leur comportement en constante évolution.

Le Cloud Computing véhiculaire peut être défini comme suit :

« Un groupe de véhicules en grande partie autonomes dont l'informatique d'entreprise, de détection, de communication et les ressources physiques peut être coordonné et allouée dynamiquement aux utilisateurs autorisés. »

2- Architecture cloud computing véhiculaire :

L'architecture de cloud computing véhiculaire repose sur trois couches: à l'intérieur de véhicules, la communication et de nuages. Comme illustré sur la Figure II-9 , la première couche est la couche de l' intérieur du véhicule, qui est responsable de la surveillance de la santé et de l' humeur du conducteur et la collecte d' informations à l' intérieur de la voiture, comme la pression et la température à l'aide de capteurs corporels, des capteurs d' environnement, des capteurs intelligents, les capteurs internes du véhicule , des capteurs de navigation par inertie (INS) et la reconnaissance du comportement du conducteur pour prédire les réflexes et les intentions du conducteur. Ensuite, les informations collectées par les capteurs doivent être envoyés vers le nuage pour le stockage ou pour être utilisé comme entrée pour différents logiciels dans la couche d'application, par exemple, propose des applications de reconnaissance et de santé environnementale. Nous partons du principe que chaque véhicule est muni d'un OBU qui comprend un système de navigation intégré, avec une carte et l'emplacement d'un UAR.

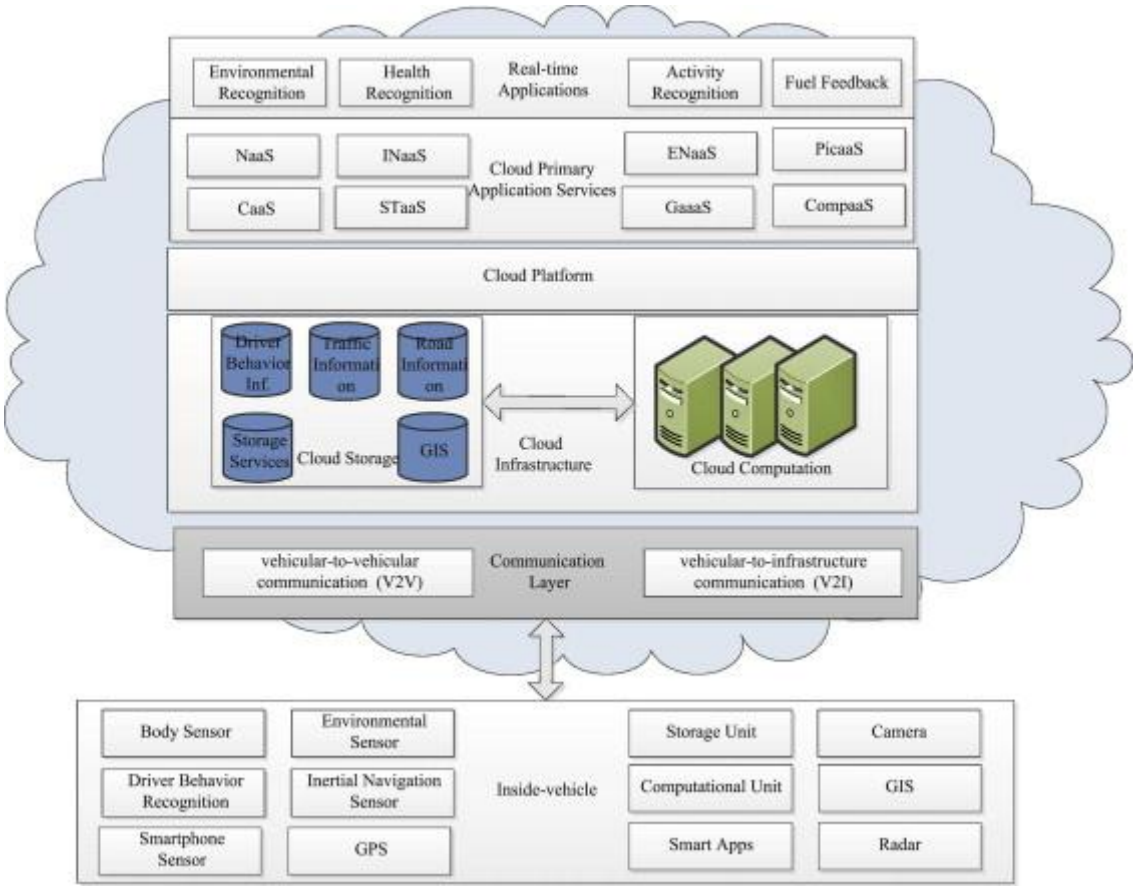


Figure II.9 : architecture de cloud computing véhiculaire

La couche suivante de cette architecture est appelée communication, qui comprend deux parties: les systèmes de véhicules à véhicule (V2V) via DSRC .Si le conducteur indique le comportement anormal sur la route tels que: un changement de direction de façon spectaculaire, la conduite sur la limite de vitesse ou l'apparition d'une défaillance mécanique importante dans le véhicule, une transmission de messages d'alerte d'urgence (EWMS) sont générés et envoyés au stockage en nuage et les véhicules environnant, qui contient l'emplacement géographique, la vitesse, l' accélération et la direction de déplacement du contrevenant. Le second composant de la couche de communication est un véhicule à infrastructure (V2I), qui est responsable de l'échange des données opérationnelles parmi les véhicules, les infrastructures et le nuage sur des réseaux sans fil tels que 3 G, par satellite ou par Internet.

L'un des avantages les plus importants de l'association est CCV de données en utilisant le stockage des Cloud, où divers organismes gouvernementaux et privés, en particulier la police ou le service de météorologie peuvent utiliser les données stockées dans le Cloud pour effectuer diverses études. Cependant, le Cloud qui est la dernière couche de l'architecture CCV, peut calculer les calculs massifs et complexes en un minimum de temps. La couche de Cloud se compose de trois couches internes: application, l'infrastructure de Cloud, et la plate-forme de Cloud. Dans la couche d'application, diverses applications et services sont considérés comme des services en temps réel ou des services primaires des Cloud, qui sont

accessibles à distance par les conducteurs, comme la rétroaction de carburant, la reconnaissance de l'activité humaine, la reconnaissance de la santé et de la reconnaissance de l'environnement. la reconnaissance de l'activité humaine est utilisée pour une analyse automatisée (ou interprétation) des événements en cours et leur contexte de données vidéo. Dans les services de base, plusieurs services sont déployés, comme le Réseau en tant que service (NaaS), stockage en tant que service (Staas), la coopération en tant que service (CaaS), l'information en tant que service (INaaS) et du divertissement en tant que service (ENaaS),. L'infrastructure Cloud se compose de deux parties,: stockage en Cloud et Cloud computing. Les données collectées par la couche de l'intérieur du véhicule sont enregistrées dans le système d'information géographique (SIG), un dispositif de contrôle du trafic routier ou un système de stockage en fonction du type d'applications. La partie de calcul est utilisée pour calculer les tâches de calcul qui fournit des performances plus rapides.

3- Formation de la perspective de l'infrastructure VC :

Dans ce paragraphe, plusieurs scénarios de formation VC sont décrits.

1- Formation VC stationnaire :

Dans plusieurs cas, un VC peut agir comme un cloud classique normal en particulier dans des environnements statiques. Laissez- nous, considérons une petite entreprise qui embauche des gens et de se concentrer sur la fourniture de services informatiques et de soutien. En permettant le covoiturage, nous aurons beaucoup de voitures garées dans le parking de l'entreprise. Toute la journée, ces voitures reste hors des ressources informatiques. En fournissant les récompenses nécessaires, la société peut demander activement la formation d'un VC stationnaire pour son personnel, qui louera les ressources. Les ressources de stockage les VC accumulera statique combinée et la puissance de calcul des véhicules participants et forment une grappe d'ordinateurs et un centre de stockage de données gigantesque distribués, qui , avec des garanties de sécurité appropriées, peut devenir un atout important pour toute entreprise d' entreprise .

2- Lié avec une infrastructure fixe :

La création d'un VC peut évoluer dans une zone instrumentée et être déployé par une certaine forme d'une infrastructure statique qui prend en charge la gestion des divers événements. Dans les zones urbaines, ces infrastructures contient des caméras, des feux de circulation, et les poteaux électriques ou l'éclairage public. Dans les routes, l'infrastructure statique contient les unités latérales de route, Détecteurs boucle inductive (ILD) et d'autres ITS matériel déployé dans la surveillance et la gestion du trafic.

3- Formation dynamique

Le soutien architectural de la formation de ce type de VC impliquera les éléments suivants. Un courtier élu parmi les véhicules essaiera de former un VC. Ensuite, le courtier obtiendra l'autorisation initiale de l'autorité pour former un VC. Parmi les véhicules, on obtenir une autorisation et de réussir, et les autres formulera la coordination et aider à former le VC. Un courtier unique, invitera les véhicules pour la formation VC dans la région

après avoir reçu l'autorisation. Cependant, les réponses des voitures vraiment avoir une base autonome. Lorsqu'un nombre suffisant de voitures sont en place, le courtier décidera d'annoncer la formation VC. Enfin, le VC accumule des ressources informatiques pour former une grande entité informatique similaire à un super - ordinateur. En utilisant une carte numérique de la région, le VC demandera l'autorité d'approbation du plan proposé, puis la mettre en œuvre. Après avoir accepté la mise en œuvre et la proposition, le VC est dissous. Dans ce scénario, il est nécessaire de réorganiser les feux de circulation dans une grande surface et donc motiver la formation de plusieurs VCs.

4. Conclusion :

Le CCV peut être le prochain paradigme de changement technologique qui fournit des solutions technologiquement réalisables et économiquement viables en convergence de réseaux de véhicules intelligents vers des systèmes autonomes de trafic, de contrôle de véhicules et de perception.

Dans le chapitre suivant on va faire une implémentation un des algorithmes de chiffrement homomorphe Paillier et on va faire des opérations mathématiques pour tester l'homomorphisme de se algorithme

chapitre III:

IMPLIMENTATION

1. Introduction : [2]

Le fonctionnement du chiffrement homomorphe, c'est un échange de données en toute confidentialité (c.à.d. les données sont chiffrées en local par l'entreprise cliente, et l'application génératrice des clés et responsable du chiffrement et déchiffrement s'exécute sur les ordinateurs de l'entreprise).

La base des données étant stockée toute chiffrée dans le Cloud, l'entreprise envoie une requête pour effectuer une opération ou un traitement et cela dans le but de bénéficier de la capacité de stockage et de calcul des serveurs Cloud. Ensuite, dès la réception du résultat de l'opération (chiffré toujours) l'application cliente le déchiffre et retourne le même résultat comme si l'opération a été effectuée sur les données en clair. La disponibilité et la confidentialité sont assurées dans ce système de chiffrement homomorphe, à condition de protéger la clé de déchiffrement qui est le seul moyen d'accéder aux données entreposées dans les serveurs cloud. Parmi les entreprises clientes qui peuvent adopter ce type d'utilisation de Cloud, sont les VANETs ou tout autre établissement financier capable de déléguer le traitement et le calcul à des serveurs distants (Cloud Computing). **La Figure III.1** ci-dessous éclaircit l'échange homomorphe des données entre l'entreprise cliente et le serveur Cloud.

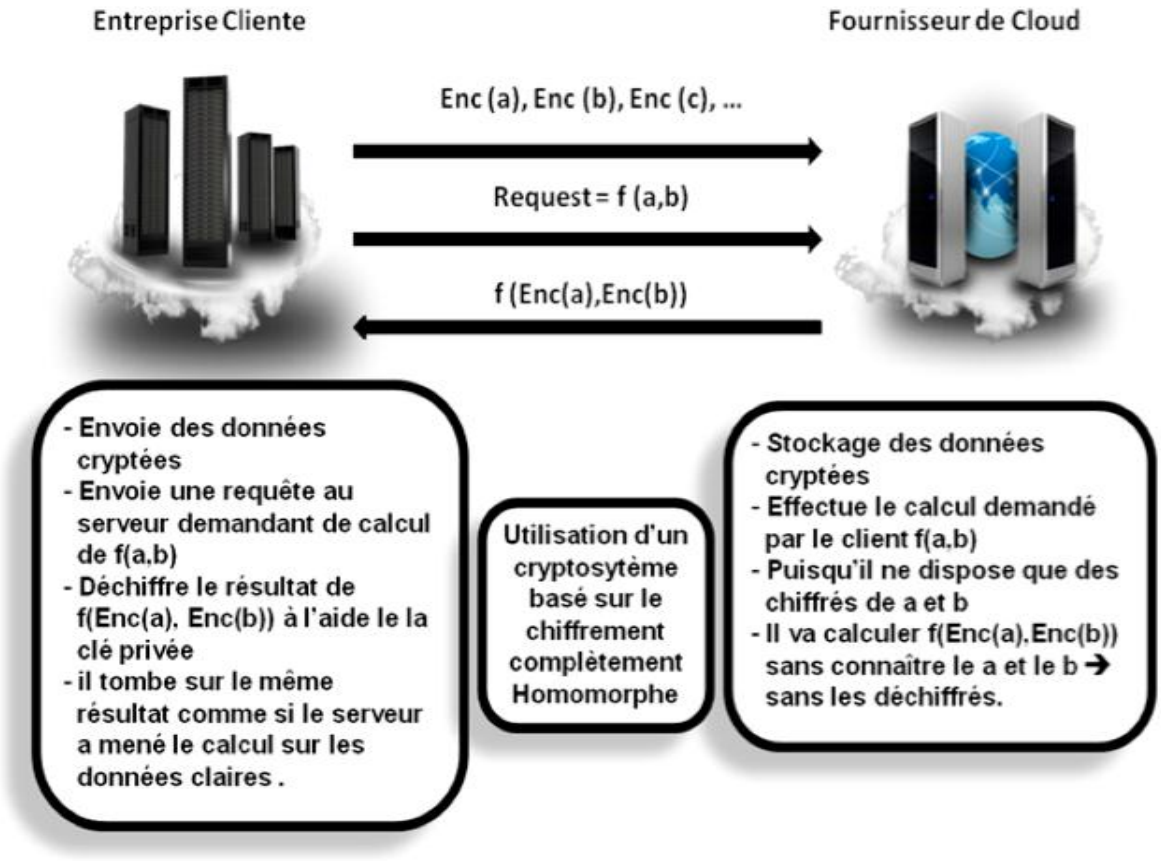


Figure III- 1 : Application du Chiffrement Homomorphe au Cloud Computing

2. Le but de l'application :

Ce travaille consiste à implémenter l'algorithme de chiffrement homomorphe Paillier en java.

3. Logiciel et langage :

a. NetBeans :

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développent Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme**. [15]**

b.JAVA :

Java est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java**.**

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante,

etc.).

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java Runtime Environment - Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes. **[16]**

Dans notre application nous avons implémenté deux classes la classe d'interface et la classe de l'algorithme Paillier

4. l'organigramme :

Voici l'organigramme qui présente les plus importantes étapes de fonctionnement de l'application :

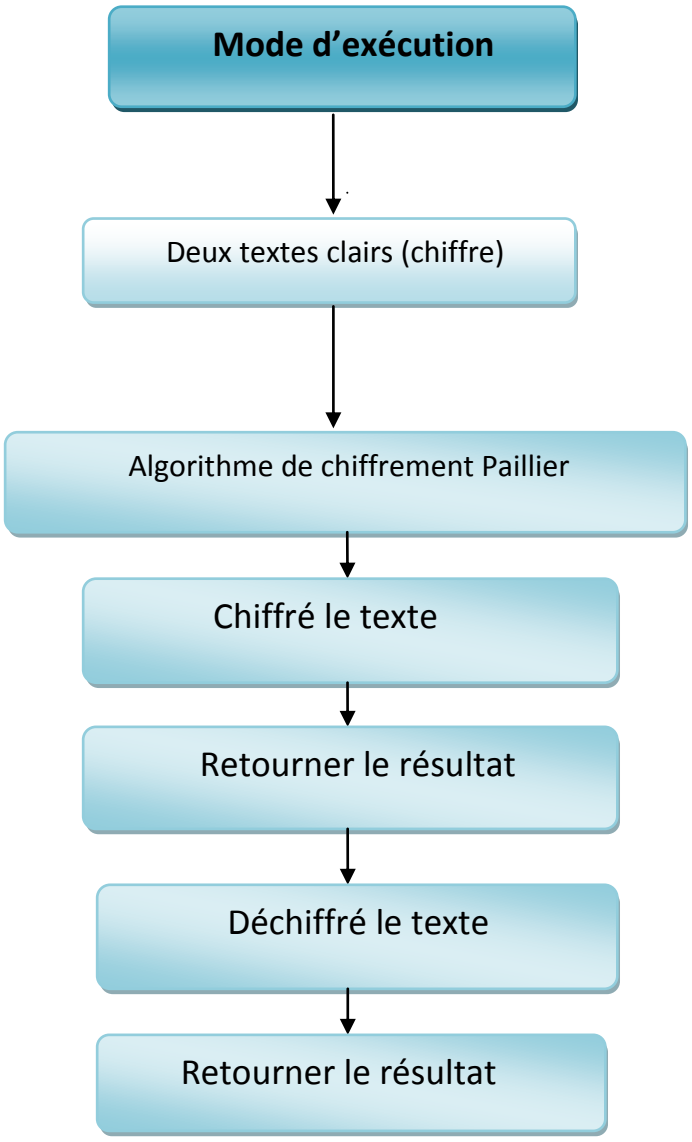


Figure III- 2 : organigramme de chiffrement et déchiffrement d'un texte

5. l'exécution de l'application :

- **L'interface de l'application :**
  La description de l'application (interface et composants) : la fenêtre d'interface de notre application est présentée ci-dessus

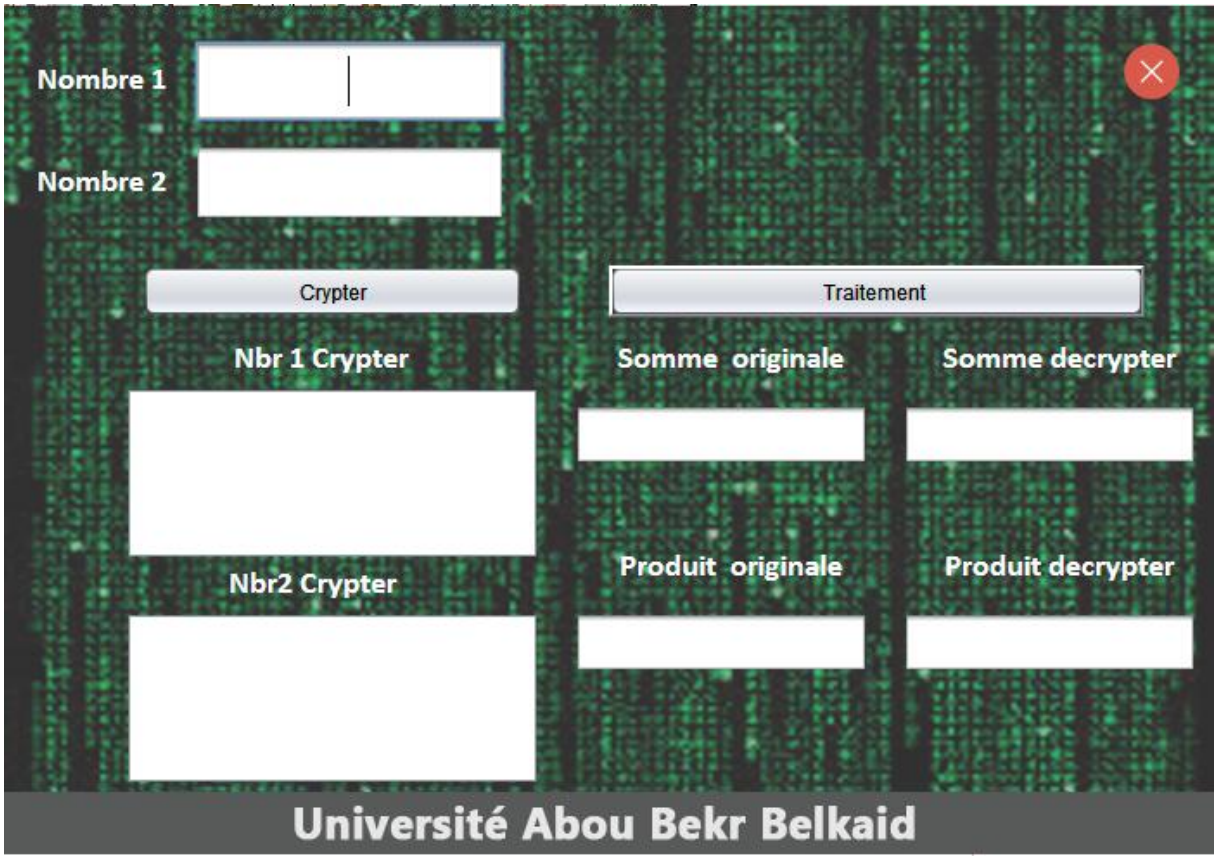


Figure III.3 : la fenêtre d'interface

Les composants :

- Deux JTextField pour saisir les nombre.
- six TextAria
  - Deux pour afficher le cryptage des deux nombre que nous avons saisie
  - Un pour afficher le résultat de la somme originale.
  - Un pour afficher le résultat de la somme décrypté.
  - Un pour afficher le résultat de produit originale.
  - Un pour afficher le résultat de produit décrypté.

- Un bouton pour faire exit.
- Un bouton pour crypter les deux nombres saisis.
- Un bouton pour le traitement.

➢ **On va faire des opérations sur des textes chiffre par Paillier :**

On obtient le résultat suivant :

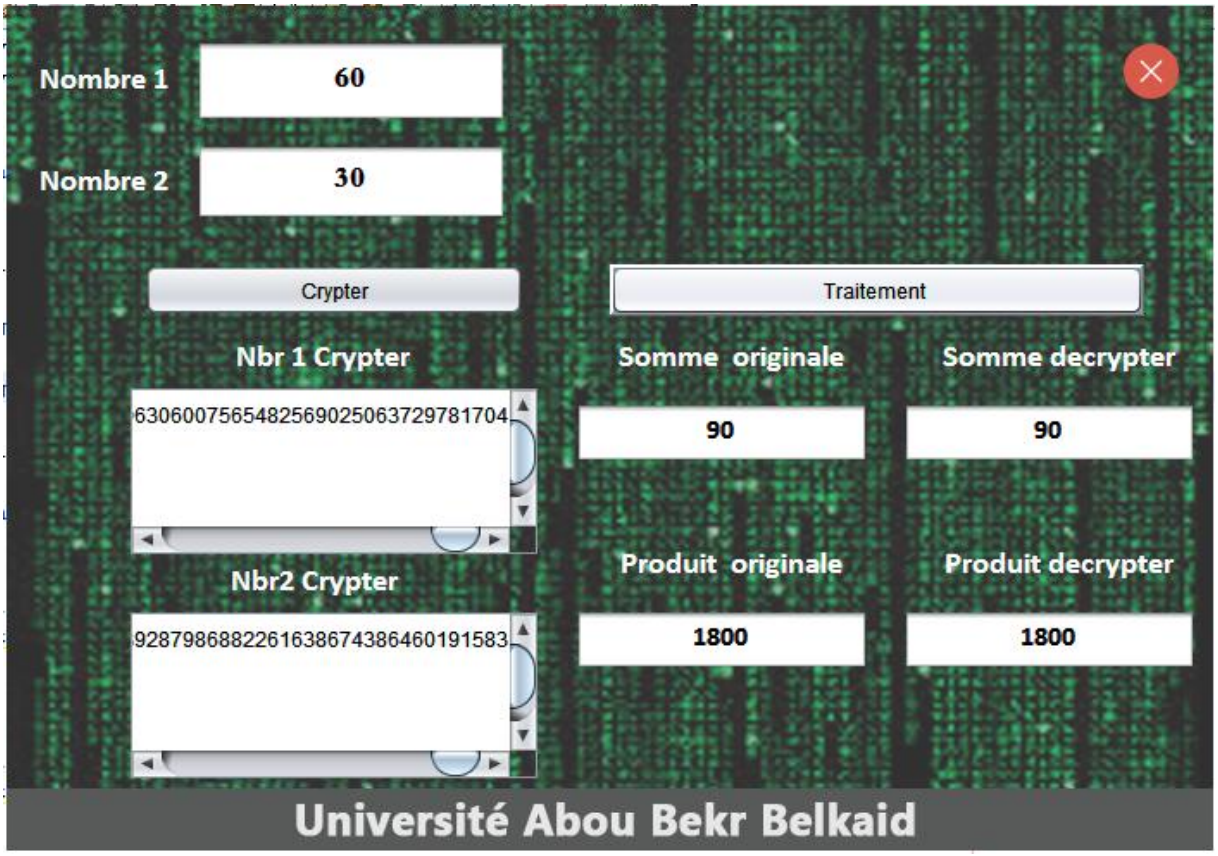


Figure III.4 : opération sur texte chiffre par Paillier

On observe que :

Paillier démontre la propriété homomorphe multiplication homomorphe additive (complètement homomorphe) :

On a Nombre1= 60 et Nombre2=30

Alors : NB1*NB2=60*30=1800

Et : NB1+NB2=60+30=90

Et :

C1=NB1crypter

C2=NB2crypter

Donc Décrypte ((C1*C2) mod n^2)=90=NB1+NB2.

Décrypte ((C1^NB2) mod n^2)=1800=NB1*NB2.

L'algorithme de Paillier faire les opérations somme et produit sur des nombres chiffré et donne les résultats sans les déchiffré.

6. Conclusion :

Dans ce chapitre on va implémenter et tester les opérations homomorphiques de l'algorithme Paillier complètement homomorphe.

L'application du chiffrement homomorphe constitue une brique importante dans la sécurité du Cloud Computing véhiculaire. Grâce à ce type de chiffrement, on pourrait sous-traiter des calculs sur des données confidentielles à des serveurs situés dans le Cloud en gardant la clé secrète qui permet de déchiffrer le résultat du calcul.

Conclusion générale

Il existe une multitude de mécanismes cryptographiques qui permettent de protéger le stockage et le traitement des données dans les environnements cloud véhiculaire. Suivant le niveau de sécurité voulu, alors différentes techniques peuvent être déployées ; chiffrement classique, preuve de stockage ou cloud Security gateway pour le stockage ; chiffrement homomorphique calcul vérifiable ou encore cloud Security gateway pour le traitement.

Aucune solution n'est encore parfaite, et la recherche académique doit encore perfectionner toutes ces techniques pour qu'elles soient davantage infaillibles. En particulier, le développement et l'implémentation pratique du chiffrement homomorphique est attendu au tournant, car cette technique peut amener une sécurité et une facilité de calcul difficilement égalable. Finalement, même si aucune solution n'est encore parfaite, la plupart sont envisageables pour protéger les données dans le cloud, à condition que les clés cryptographiques utilisées soient sous le contrôle de l'utilisateur.

.Bibliographie

[1]: B.KADRI, << initiation à la cryptographie >>, cours 2017

[2]: Maha Tebaa <<chiffrement Homomorphe appliqué au cloud bancaire>>, thèse de doctorat informatique, Université Mohammed V, Faculté des sciences Rabat soutenue le 12/11/2015

[3]: Paillier P, <<Public-key cryptosystems based on composite degree residuosity classes>>, *Advances in Cryptology Eurocrypt*, 1592:223–238, 1999.

[4]: Goldwasser S. and Micali S, << Probabilistic encryption>>, *Journal of computer and System sciences*, 28(2):270–299, 1984.

[5]: Rivest R. L. Adleman L. and Dertouzos M. L., <<On data banks and privacy Homomorphism's">>, *Foundations of secure computation*, 4(11):169–180, 1978.

[6]: EL Gamal T. <<A public key cryptosystem and a signature scheme based on discrete logarithms>>, *Advances in cryptology, Springer Berlin Heidelberg*, 1985.

[7]: Gentry C, << Fully homomorphic encryption using ideal lattices>>, *STOC* 9:169–178, 2009.

[8] : Mounet J. *Le livre blanc du Cloud Computing, tout ce que vous devez savoir sur l'informatique dans le nuage*. Syntec informatique, www.syntec-informatique.fr, 2ème Trimestre 2010.

[9]: Marinos A. and Briscoe G. Community cloud computing. *In Cloud Computing, Springer Berlin Heidelberg*, pages 472–484, 2009.

[10] : Source PAC. *ZDNet.fr/chiffres-clés*. 2014.

[11] : Md. Whaiduzzamanetal. <<A survey on vehicular cloud computing >> (2014)325–344

[12] : Azdad Nabila, << La sécurité dans les réseaux vanet's, étude de cas du protocole css-olsr>>, Projet fin d'étude master science et technique, systèmes intelligents et réseaux, Université Sidi Mohamed Ben Abdellah Fes 2015-2016.

[13] : Ayoub Benchaban, Ramla Bensaci <<Analyse des protocoles de routages dans les réseaux vanet's >>, Mémoire master académique en Informatique Industriel, Université Kasdi Merbah- Ouargla 2013-2014.

[14] : Rabah MERAIHI, Sidi-Mohammed SENOUCI, Djamal-Eddine MEDDOUR et Moez JERBI, << Chapitre communications véhicule à véhicule applications et perspectives>>,2006.

[15] : **http://fr.wikipedia.org/wiki/NetBeans** consulter en juin 2017

[16] : **http://ipeti.forumpro.fr/t21-definition-de-langage-java-java-script** consulté en juin 2017

Glossaire

A

A-STAR: Anchor-based Street and Traffic Aware Routing

AODV: Ad hoc On-demand Distance Vector

B

BRP: Bordercast Resolution Protocol

C

CaaS : coopération en tant que service

Cc : Computing-Cloud

CCV : Véhiculaire Cloud Computing

D

DSDV : Destination-Sequenced Distance-Vector

DSR : Dynamic Source Routing

E

ENaaS : Du divertissement en tant que service

G

GPSR: Greedy Perimeter Stateless Routing

GLS: Grid Location Service

GyTAR: Greedy Traffic-Aware Routing protocol

GPS: Global Position System

GSR: Global State Routing

GM: Goldwasser-Micalli

I

IARP: IntrAzone Routing Protocol

IEEE: Institute of Electrical and Electronics Engineers

IERP: IntErzone Routing Protocol

IETF : Internet Engineering Task Force

INaaS : L'information en tant que service

INRIA : Institut National de la Recherche en Informatique et Automatique

ITS : Systèmes de Transport Intelligents.

M

Mcc: Mobile Computing-Cloud

MORA: *MOvement-based Routing Algorithm*

MPR: Multi-Point Relaying

N

NIST: National Iinstitut of Standard and Technology

NaaS : comme le Réseau en tant que service

NDP: NeighbourDiscovery Protocol

O

OBU: On-Board Unit

OLSR: Optimized Link State Routing

R

RFC: Request for Comment

RERR: Route Error Message

RREP: Route Reply Message

RREQ: Route Request Message

RSA: Ronald Rivest, Adi Shamir et Leonard Adleman

RSU: Road Side Unit

S

STI : les systèmes de transport intelligents

Staas : stockage en tant que service

SIG : système d'information géographique

U

UAR:

UMB: Urban Multi hop Broadcast

V

VADD: *Vehicle-Assisted Data Delivery*

VANET: Vehicular Ad Hoc Network

V2V: Véhicule à véhicule via DSRC

V2I: Un véhicule à infrastructure

Z

ZRP: Zone Routing Protocol

La norme 802.11 : est un ensemble de normes concernant les réseaux sans fil locaux (le Wi-Fi)

Annex

```
import java.awt.Color;
import java.math.BigInteger;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author mohamed
 */
public class paillier_frame extends javax.swing.JFrame {

    /**
     * Creates new form paillier_frame
     */
    public paillier_frame() {
        setUndecorated(true);
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
```

```
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    nb1 = new javax.swing.JTextField();
    nb2 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    so = new javax.swing.JTextField();
    sd = new javax.swing.JTextField();
    jButton2 = new javax.swing.JButton();
    err = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    cry = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    decry = new javax.swing.JTextArea();
    jButton3 = new javax.swing.JButton();
    pd = new javax.swing.JTextField();
    po = new javax.swing.JTextField();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
```

```
        setMinimumSize(new java.awt.Dimension(700, 490));
        getContentPane().setLayout(null);

        jLabel2.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(255, 255, 255));
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setText("Nombre 1");
        getContentPane().add(jLabel2);
        jLabel2.setBounds(10, 20, 92, 47);

        jLabel3.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(255, 255, 255));
        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel3.setText("Nombre 2");
        getContentPane().add(jLabel3);
        jLabel3.setBounds(10, 80, 92, 44);

        nb1.setFont(new java.awt.Font("Traditional Arabic", 1, 18)); // NOI18N
        nb1.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        getContentPane().add(nb1);
        nb1.setBounds(110, 20, 180, 47);

        nb2.setFont(new java.awt.Font("Traditional Arabic", 1, 18)); // NOI18N
        nb2.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        getContentPane().add(nb2);
        nb2.setBounds(110, 80, 180, 44);

        jButton1.setText("Traitement ");

jButton1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.R
AISED));
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1);
jButton1.setBounds(350, 150, 310, 30);

jLabel1.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Somme  originale");
getContentPane().add(jLabel1);
jLabel1.setBounds(340, 190, 160, 29);

jLabel4.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("Somme decrypter");
getContentPane().add(jLabel4);
jLabel4.setBounds(530, 190, 160, 29);

jLabel5.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Nbr 1 Crypter ");
getContentPane().add(jLabel5);
jLabel5.setBounds(120, 190, 130, 28);

jLabel6.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
```

```
        jLabel6.setForeground(new java.awt.Color(255, 255, 255));
        jLabel6.setText("Nbr2 Crypter");
        getContentPane().add(jLabel6);
        jLabel6.setBounds(130, 320, 106, 29);

        so.setFont(new java.awt.Font("Calibri", 1, 16)); // NOI18N
        so.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        getContentPane().add(so);
        so.setBounds(330, 230, 170, 35);

        sd.setFont(new java.awt.Font("Calibri", 1, 16)); // NOI18N
        sd.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        getContentPane().add(sd);
        sd.setBounds(520, 230, 170, 35);

        jButton2.setBackground(new Color(0,0,0,0));
        jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/error (1).png"))); //
NOI18N
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2);
        jButton2.setBounds(640, 20, 48, 36);

        err.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
        err.setForeground(new java.awt.Color(255, 255, 255));
        err.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(err);
```

```
err.setBounds(300, 80, 380, 40);

cry.setColumns(20);
cry.setRows(5);
jScrollPane1.setViewportView(cry);

getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(70, 220, 240, 100);

decry.setColumns(20);
decry.setRows(5);
jScrollPane2.setViewportView(decry);

getContentPane().add(jScrollPane2);
jScrollPane2.setBounds(70, 350, 240, 100);

jButton3.setText("Crypter");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
getContentPane().add(jButton3);
jButton3.setBounds(80, 150, 220, 30);

pd.setFont(new java.awt.Font("Calibri", 1, 16)); // NOI18N
pd.setHorizontalAlignment(javax.swing.JTextField.CENTER);
getContentPane().add(pd);
pd.setBounds(520, 350, 170, 35);
```

```
        po.setFont(new java.awt.Font("Calibri", 1, 16)); // NOI18N
        po.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        getContentPane().add(po);
        po.setBounds(330, 350, 170, 35);

        jLabel8.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
        jLabel8.setForeground(new java.awt.Color(255, 255, 255));
        jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel8.setText("Produit  originale");
        getContentPane().add(jLabel8);
        jLabel8.setBounds(340, 310, 160, 29);

        jLabel9.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N
        jLabel9.setForeground(new java.awt.Color(255, 255, 255));
        jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel9.setText("Produit decrypter");
        getContentPane().add(jLabel9);
        jLabel9.setBounds(530, 310, 160, 29);

        jLabel7.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/Sans titre-1.jpg")));
// NOI18N
        getContentPane().add(jLabel7);
        jLabel7.setBounds(0, 0, 700, 490);

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Runtime.getRuntime().exit(0);        // TODO add your handling code here:
```

```
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
err.setText("");
        if(nb1.getText().length()>0 && nb1.getText().length()>0 ){

/* instantiating an object of Paillier cryptosystem*/
Paillier paillier = new Paillier();
/* instantiating two plaintext msgs*/
BigInteger m1 = new BigInteger(nb1.getText());
BigInteger m2 = new BigInteger(nb2.getText());

/* encryption*/
BigInteger em1 = paillier.Encryption(m1);
BigInteger em2 = paillier.Encryption(m2);

BigInteger product_em1em2 = em1.multiply(em2).mod(paillier.nsquare);
BigInteger sum_m1m2 = m1.add(m2).mod(paillier.n);



so.setText(""+sum_m1m2.toString());
sd.setText(""+paillier.Decryption(product_em1em2).toString());



BigInteger expo_em1m2 = em1.modPow(m2, paillier.nsquare);
BigInteger prod_m1m2 = m1.multiply(m2).mod(paillier.n);

po.setText(""+  prod_m1m2.toString());
pd.setText(""+paillier.Decryption(expo_em1m2).toString());
    }
```

```
        else{
err.setText("Veuillez entrer 2 chiffres");
err.setBackground(new Color(88,90,90,255));
        }// TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
err.setText("");
        if(nb1.getText().length()>0 && nb1.getText().length()>0 ){

/* instantiating an object of Paillier cryptosystem*/
Paillier paillier = new Paillier();
/* instantiating two plaintext msgs*/
BigInteger m1 = new BigInteger(nb1.getText());
BigInteger m2 = new BigInteger(nb2.getText());

/* encryption*/
BigInteger em1 = paillier.Encryption(m1);
BigInteger em2 = paillier.Encryption(m2);
/* printout encrypted text*/
cry.setText(""+em1);
decry.setText(""+em2);

 }
        else{
err.setText("Veuillez entrer 2 chiffres");
err.setBackground(new Color(88,90,90,255));
        }
```

```
// TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(paillier_frame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(paillier_frame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(paillier_frame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(paillier_frame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new paillier_frame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextArea cry;
    private javax.swing.JTextArea decry;
    private javax.swing.JLabel err;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JScrollPane jScrollPane1;
```

```
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextField nb1;
    private javax.swing.JTextField nb2;
    private javax.swing.JTextField pd;
    private javax.swing.JTextField po;
    private javax.swing.JTextField sd;
    private javax.swing.JTextField so;
    // End of variables declaration
}
```