Résumé

Certaines régions présentes sur l'image sont trop sensibles et importantes ou elle contient des informations critiques, d'où le besoin de les protéger. La cryptographie était une science réservée aux militaires, elle permet de protéger les informations tout en gardant le secret pour garantir la confidentialité. Les travaux de recherche de ce mémoire s'inscrivent dans le cadre la sécurité des images satellitaires, par un algorithme asymétrique RSA, qui est fondé sur l'utilisation d'une paire de clés composée : une clé publique pour chiffrer et une clé privée pour déchiffrer les données confidentielles. L'originalité de ce mémoire consiste à proposer un cryptosystème à base d'algorithme RSA pour assurer les deux services de la sécurité l'authentification et l'intégrité, plus la technique de la programmation parallèle qui diminue le temps de chiffrer/déchiffrer l'image satellitaire.

Mots clés: Cryptographie, Image satellitaire, Programmation parallèle, RSA.

Abstract

Certain parts present in the image are too sensitive, important or it contains critical information, hence the need to protect. The Cryptography was a science reserved for the military, it helps protect the information while keeping it secret to ensure confidentiality. The research of this memoir is in the context of the security of satellite images, by the asymmetric algorithm called RSA, which is based on the use of a pair of composite keys: a public key to encrypt and a private key to decrypt the confidential data. The originality of this memoir is to provide a cryptosystem based on the RSA algorithm to ensure both the security services, authentication and integrity, plus the technique of parallel programming that reduces the time to encrypt/decrypt a satellite image.

Keywords: Cryptography, Satellite image, parallel Programming, ,RSA.

Sommaire

Chapitre IV : Application et résultat

Listes des abréviations

AES : Advanced Encryption Standard

DES : Data Encryption Standard

NIST : National Institute of Standards and Technology

RSA : Ron Rivest, Adi Shamir, Len Adleman

CPU : Central Processing Unit

CNA : Convertisseur Numérique Analogique

SIMD : Single Instruction Multiple Data

MIMD : Multiple Instruction Multiple Data

SISD : Single Instruction Single Data

MISD : Multiple Instruction Single Data

SMP : Symmetric Multiprocessing

DAG : Directed Acyclic Graph

JADE : Java Agent Development Environment

UMA : Uniform Memory Access

NUMA : Non Uniform Memory Access

GPU : Graphics Processing Unit

OpenMP : Open Multi-Processing

OpenCL : Open Computer Language

API : Application Programme Interface

MPI : Message Passing Interface

RAM : Random Access Memory

Listes des figures

Chapitre I : La cryptographie

Chapitre II : Le parallélisme

Listes des figures

Listes des tableaux

Chapitre I : La cryptographie

Introduction général

Depuis le début de la civilisation, progrès et développements de la technologie, le besoin de cacher et de dissimuler préoccupe l'humanité. La confidentialité apparaissait notamment nécessaire lors des luttes pour l'accès au pouvoir. Puis elle à été énormément développée pour les besoins militaires et diplomatiques.

Aujourd'hui, de plus en plus d'application dites civiles nécessitent la sécurité des données transitant entre deux interlocuteurs via un vecteur d'information comme les réseaux de télécommunications actuels et futurs. Ainsi les banque l'utilisent pour assurer la confidentialité des opérations avec leurs clients, les laboratoires de recherche s'en servent pour l'échange des informations dans le cadre d'un projet d'étude commun, les chefs militaires pour donner leurs ordre de bataille, …etc.

La cryptographie ou l'art de chiffrement étant un sujet très vaste, permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible; c'est ce qu'on appelle le chiffrement, qui à partir d'un texte en clair, donne un texte chiffré ou cryptogramme. Inversement le déchiffrement est l'action qui permet de reconstruire le texte en claire à partir d'un texte chiffré.

La cryptographie moderne, les transformations en question sont des fonctions mathématique, appelées algorithmes cryptographiques, qui dépendent d'un paramètre appelé clef.

Puisque on ait dans l'air de la technologie, le temps joue un rôle très essentiel dans notre vie, et cela doit comprendre comment l'exploiter et l'utiliser. Sachant que l'algorithme de chiffrement et de déchiffrement demande une durée un peut long se qui nous a amené à utiliser une méthode de diminution du temps qui est la programmation parallèle.

Le calcule parallèle ou parallélisme est l'ensemble des méthodes qui permettent d'organiser une série de calcule sur un ensemble d'unités de calcule concurrentes, comporte plusieurs étapes : il faut trouver le parallélisme, puis découper le problème en tâche. Il faut ensuite définir un ordre d'exécution et associer à chacune des tâche un processeur chargé de leur exécution.

Le travail réalisé dans ce mémoire se situe dans le cadre des approches qui proposent

Introduction général

Le deuxième chapitre, aborde la notion de parallélisme, modèle, et les différentes architectures de parallélisme, et les principaux modèles de programmation parallèle.

Le troisième chapitre, comporte les méthodes utilisées qui est une présentation théorique d'algorithme de chiffrement et de déchiffrement RSA avec la machine de mémoire partagé de la bibliothèque OpenMP de programmation parallèle.

Le dernier chapitre sera consacré à la réalisation et l'implémentation de notre application pour l'algorithme de chiffrement et de déchiffrement RSA d'une image satellitaire avec la machine à mémoire partagée pour la programmation parallèle avec une explication détaillée, leur sécurité et leur performance sont analysées et évaluées.

Nous terminons ce mémoire par une conclusion générale qui présente le bilan du travail réalisé et les perspectives envisagées.

Chapitre I : La cryptographie

I.1 Introduction

Dans ce chapitre, nous avons présenté certaines définitions importantes, une terminologie qui faciliteront la compréhension des concepts et des objectifs des travaux de recherche développés dans les sections suivantes.

La cryptographie a évolué en trois périodes historiques :

- **La cryptographie mécanique** : Il s'agit de la cryptographie qui utilise des moyens mécaniques pour chiffrer un message. Cette cryptographie s'étend de l'antiquité jusqu'à la fin de la seconde guerre mondiale environ. De nos jours, elle n'a plus cours.
- **La cryptographie mathématique** : Il s'agit de la cryptographie qui utilise les mathématiques pour chiffrer un message. Cette cryptographie a commencé aux environs de la fin de la deuxième guerre mondiale et c'est celle que l'on utilise de nos jours.
- **La cryptographie quantique** : Il s'agit de la cryptographie dont les bases reposent sur la physique quantique. Nous sommes en train de la voir émerger de nos jours et nul doute qu'elle ne remplace dans les années qui viennent la cryptographie basée sur les mathématiques **[1]**.

Avec le développement rapide des technologies et des applications spatiales, plusieurs images satellitaires sont prises. Certaines images doivent être chiffrées avant de les transmettre au sol. Nous parlerons dans cette phase sur la cryptographie des images satellite et leurs définitions, les différents types des images.

I.2 Terminologie

On suppose que quelqu'un veut envoyer un message à un destinataire, et veut être sûr que personne d'autre ne peut lire le message. Cependant, il y a la possibilité que quelqu'un d'autre ouvre la lettre ou intercepte la communication électronique.

Dans la terminologie cryptographique, le message est appelé plain-texte ou clair-texte. Chiffrer le contenu du message dans un tel chemin qui cache son contenu aux étrangers est appelé le cryptage. Le message codé est appelé le chiffre-texte. Le processus d'extraire le plain-texte du chiffre-texte est appelé décryptage. Le cryptage et le décryptage ion font habituellement usage d'une clef.

Chapitre I : La cryptographie

I.3 Définition de la cryptographie

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 **[3].**

I.3.1 Objectif

L'objectif fondamental de la cryptographie est de permettre à deux personnes de communiquer au travers d'un canal peu sûr (téléphone, réseau informatique ou autre) sans qu'un opposant puisse comprendre ce qui est échangé **[4]**.

Globalement, la cryptographie permet de résoudre cinq problèmes différents :

- **La confidentialité:** qui consiste à rendre l'information inintelligible à tous ceux qui pourraient intercepter le message.
- **Le contrôle d'accès:** qui permet de limiter l'accès aux données, serveurs aux personnes autorisées (mot de passe Unix, par exemple).
- **L'intégrité des données:** qui consiste à vérifier que cette donnée n'a pas été altérée frauduleusement.
- **L'identification :** qui permet d'assurer de l'authentification des partenaires et de l'origine des messages.
- **La non répudiation :** pour que les partenaires ne puisse nier le contenu des informations **[5]**.

I.3.2 Principe

On s'intéressant un peu à la cryptographie, on a entendu parler des principes de Kerckhoffs, en particulier celui affirmant que toute méthode de chiffrement est connue de l'ennemi et que la sécurité du système ne dépend que du choix des clés. Bien sûr, cela n'implique pas nécessairement que la méthode doit être publique, mais simplement qu'elle doit être considérée comme publique lors de son invention **[6]**.

Le chiffrement permet l'échange sûr des renseignements privés et confidentiels. Un texte

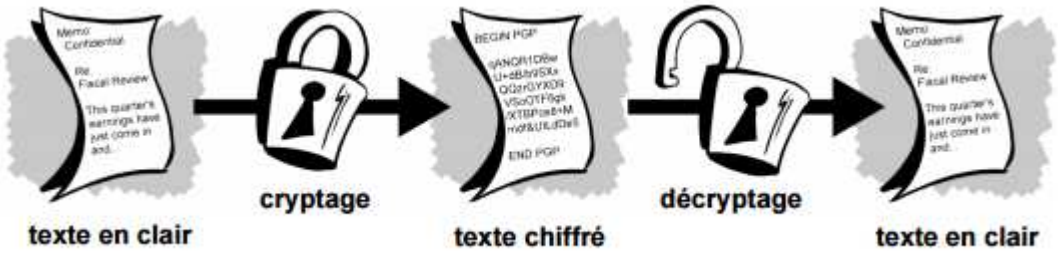


Fig.I.01: Cryptage et décryptage **[8]**.

I.3.3 Utilisation

Elle constitue la base de la plupart des techniques de sécurité:

- Echanges sécurisés sur Internet
- Confidentialité des transactions bancaires
- Protection des secrets industriels et commerciaux
- Protection du secret médical
- Protection des systèmes informatiques contre les intrusions
- Par le biais de la signature électronique:
  - Identification des correspondants
  - Non répudiation des transactions
  - Garantie d'intégrité des documents **[9]**.

I.3.4 Méthode de la cryptographie

I.3.4.1 Cryptographie dans l'Antiquité

La cryptographie est l´étude des messages secrets. Le terme cryptographie vient en effet des mots grecs anciens

A. La technique du rouleau assyrien

Le premier exemple indéniable de cryptographie remonte au moins au V^{e} siècle avant notre

Fig.I.02 : Scytale **[5]**.

B. Carré de Polybius

Le premier texte connu, traitant explicitement de cryptographie, semble être le traité de Aeneas Tacticus sur la « Défense des fortifications ». On sait aussi qu'un autre grec, Polybius développa un système de codage des lettres de l'alphabet par des paires de symboles, utilisant ce qu'on appelle un « carré de Polybius ». Son idée a souvent été reprise par la suite. L'utilisation du carré de Polybius consiste à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne ou se trouve cette lettre **[10]**.

Fig.I.03 : carré polybius**[10]**.

C. Système de César (par décalage)

En 44 avant notre ère, Jules César utilisait une simple méthode de substitution de lettres pour

I.3.4.2 Cryptographie classique

Le Moyen Âge représente une période charnière pour le développement des technologies cryptographiques. Ce fut en effet l'époque où les chiffrements classiques furent décodés, engendrant par là même l'apparition de nouvelles méthodes. Dans le même temps, l'intensification des activités diplomatiques entraîna un accroissement du volume d'informations confidentielles échangées, et donc de l'usage de la cryptographie **[11]**.

A. Procédé de Vigenère

C'est une méthode de chiffrement est le fruit du travail de Blaise de Vigenère. Il semble qu'elle fut mise au point par Vigenère lors de ses visites au Vatican. Le principe à la base de cette méthode consiste à utiliser une différente substitution alphabétique à chaque position, ce qui rend l'analyse des fréquences un peu moins attrayante **[12]**.

Dans l'exemple qui suit, le mot-clé est MONTREAL. Pour chiffrer, on choisit la rangée de la table 1 ci-dessous qui correspond à la lettre appropriée du mot-clé et on opère une substitution alphabétique avec la lettre située à l'intersection de la colonne correspondant à celle-ci et de la rangée correspondant à la lettre du texte en clair. Le chiffrement du texte en clair s'effectue donc par autant de substitutions différentes qu'il y a de lettres dans le mot-clé.

Mot clé : MONTREALMONTREALMO

Texte en clair : CHARDASSAUTADROITE

Texte chiffré : OVNKUESDMIGTUVOTFS

Pour cette méthode, le destinataire doit connaître le mot-clé et la table de chiffrement. Cette table peut être aussi simple que celle présentée ci-dessous. Le déchiffrement est accompli en procédant à l'inverse, tout simplement **[12]**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N

seigneur de guerre Uesugi Kenshin, la création d'une table de cryptage à partir d'un carré de Polybe. L'alphabet japonais traditionnel (tiré du poème iroha-uta) comportant 48 lettres, la table se compose de sept lignes et sept colonnes, chacune désignée par un numéro. Dans le message crypté, chaque lettre sera alors représentée par un numéro à deux chiffres **[11]**.

7	6	5	4	3	2	1	
we	a	ya	ra	yo	chi	i	1
hi	sa	ma	mu	ta	ri	ro	2
mo	ki	ke	u	re	nu	ha	3
se	yu	fu	wi	so	ru	ni	4
su	me	ko	no	tsu	wo	ho	5
n	mi	e	o	ne	wa	he	6
	shi	te	ku	na	ka	to	7

Table.I.02: table de conversion **[11]**.

I.3.4.3 Cryptographie moderne

Il existe deux techniques principales permettant de crypter les informations : le cryptage symétrique (également appelé cryptage de clé secrète) et le cryptage asymétrique (également appelé cryptage de clé publique.)

A. Cryptographie symétrique

Le premier type de cryptage à avoir été inventé, bien avant notre ère, l'émetteur et le récepteur commencent par se mettre d'accord sur une méthode de cryptage. Ensuite, ils doivent s'échanger la « clé » de cryptage. Si l'algorithme est une simple permutation de lettres, la clé est le tableau de correspondance entre les lettres non cryptées et les lettres cryptées.

L'émetteur et le récepteur doivent connaître une même clé, servant à la fois aux opérations de cryptage et de décryptage. Le cryptage symétrique est généralement simple, rapide, et il peut

Le Rijndael procède par blocs de 128 bits, avec une clé de 128 bits également. Chaque bloc subit une séquence de 5 transformations répétées 10 fois : addition de la clé secrète (par un ou exclusif), transformation non linéaire d'octets, Décalage de lignes, Brouillage des colonnes, Addition de la clé de tour **[14]**.

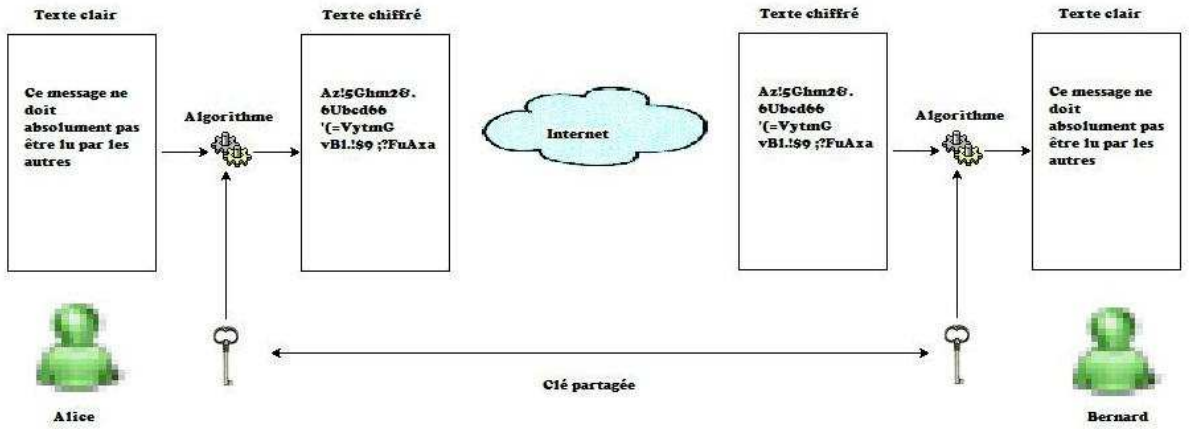


Fig.I.04 : Principe de l'algorithme symétrique **[1]**.

B. Cryptographie asymétrique

Fait intervenir non pas une clé secrète, mais deux : l'une pour le cryptage, et l'autre pour le décryptage. Chaque personne possède un jeu de deux clés : il conserve secrètement la clé de décryptage, que l'on appelle donc la clé « privée », mais il divulgue librement la clé de cryptage, que l'on appelle la clé « publique ».

Si Alice veut envoyer un message à Bernard, il commence par lui demander sa clé publique. Grâce à cette clé publique, Alice crypte son message, et envoie le résultat à Bernard. Puisque Bernard est le seul à posséder la clé privée, il sera le seul à pouvoir décrypter le message. L'avantage par rapport au cryptage symétrique est que la clé de décryptage n'est jamais échangée.

La magie du cryptage asymétrique permet à deux personnes de communiquer ensemble de façon absolument confidentielle, même si un espion peut écouter toutes leurs communications, du premier au dernier message.

Parmi les méthodes de la cryptographie asymétrique le plus utilisé RSA. L'algorithme RSA, dont le nom est composé des initiales de ses inventeurs Ron Rivest, Adi Shamir et Len

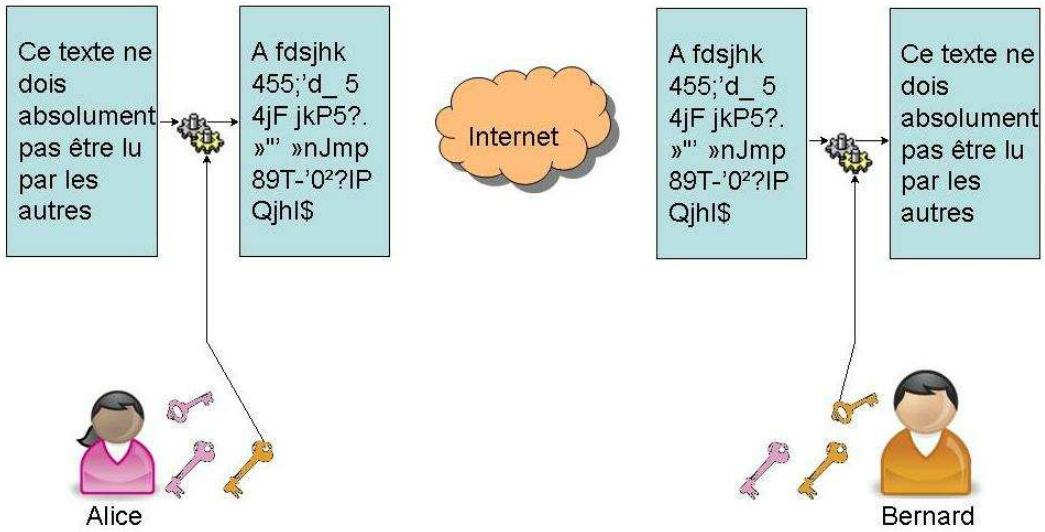


Fig.I.05 : Chiffrement avec l'algorithme asymétrique **[1]**.

Le cryptosystème d'ElGamal, ou chiffrement El Gamal est un protocole de cryptographie asymétrique inventé par Taher Elgamal en 1984 et construit à partir du problème du logarithme discret. Il possède la particularité d'être probabiliste, ce qui permet d'effectuer les preuves de sécurité sans avoir à recourir à de la complétion **[15]**.

I.3.5 Les avantages et les inconvénients du chiffrement asymétrique par rapport au chiffrement symétrique :

• Un avantage important des chiffrements asymétriques par rapport aux chiffrements symétriques est qu'aucun canal secret n'est nécessaire pour pratiquer l'échange de la clé publique. Le récepteur doit simplement être sûr de l'authenticité de la clé publique. Les chiffrements symétriques demandent un canal secret pour envoyer la clé secrète – générée d'un côté du canal de communication –vers l'autre côté

• Les chiffrements asymétriques créent aussi moins de problèmes de gestion de clés que les chiffrements symétriques. 2n clés seulement sont nécessaires pour que n entités communiquent en toute sécurité entre elles. Dans un système basé sur des chiffrements symétriques, il faudrait n(n – 1)/2 clés secrètes. Dans une entreprise de 5 000 employés, par

• Un autre inconvénient est que les chiffrements symétriques peuvent être percés par une attaque par « force brutale » : essai systématique de toutes les clés possibles jusqu'à trouver la bonne.

En raison de ces caractéristiques, les chiffrements asymétriques sont généralement utilisés pour l'authentification de données (par l'intermédiaire de signatures numériques), pour la distribution d'une clé de cryptage symétrique en masse (aussi appelée enveloppe numérique), pour des services de non-répudiation, et pour un accord de clé. Les chiffrements symétriques sont plutôt réservés au cryptage de masse **[16]**.

I.4 La cryptographie des images satellitaire

La transmission sécurisée et authentique des images satellitaires est devenue au cours de ces dernières années un sujet de très grand intérêt pour la communauté étant donné la nécessité de transfert sécurisé d'information entre les satellites et les stations terriennes de contrôle. La technique de cryptage de chiffrement d'image est un outil très performant permettant d'assurer la confidentialité des images transmises par les satellites **[17]**.

I.4.1 Images satellitaires

L'image satellitaire est une image numérique, c'est-à-dire un assemblage de pixels, ou surfaces élémentaires, référencés en ligne et colonnes formant un maillage régulier de la surface totale balayée par le capteur. Chaque pixel contient une somme d'informations codées par les valeurs des comptes radiométriques et les coordonnées en pixels.

Les images satellitaires sont le produit d'une technologie qui utilise le rayonnement réfléchir ou émis par un objet dans des intervalles de longueurs d'ondes données.

Ce rayonnement est enregistré par un capteur (caméra ou scanneur) installé à bord d'un satellite.

L'information est transmise à une station de réception terrestre, stockée puis traitée pour réaliser une image satellitaire **[18]**.

I.4.2 Différents types d'images

Il existe trois types d'images qui peuvent être stockés dans la mémoire graphique et affichés

Le deuxième type d'image qui peut être stocké dans la mémoire graphiques et affiché sur écran est appelé une image en niveau de gris. Comme l'image pseudo- couleur, l'image en niveau de gris a seulement une entrée unique (représentant un canal simple). À la différence de l'image pseudo-couleur, pour laquelle les trois convertisseurs numérique-analogique (CNA) produisent des niveaux indépendants du rouge, vert et de bleu, les CNA pour l'image en niveau de gris produisent exactement le même niveau de rouge, vert et de bleu. Rappelons que des valeurs d'intensité égale du rouge, vert et bleu forment en ensemble des nuances de gris.

Le troisième type d'image est l'image classifiée (labelled image) qui se compose des pixels dont les valeurs représentent une étiquette qui indique une propriété d'une certaine espèce. L'étiquette elle-même n'a aucune signification numérique. Il existe plusieurs méthodes de classifications pour ce type d'image. Ces méthodes permettent d'affecter chaque pixel à une catégorie spécifique, tel que : un type particulier des roches. Ces catégories sont décrites par des étiquettes comme `1', `2', ou `3', etc. qui indiquent par exemple l'eau, forêts ou sol. Pour afficher une telle image à l'écran, l'image classifiée est placée d'abord dans la mémoire et les trois CNAs (Rouge, vert et Bleu) sont programmés pour attribuer des valeurs de RGB aux différents pixels (juste comme dans le cas de l'image pseudo-couleur) **[19]**.

I.5 Conclusion

Dans ce chapitre, nous avons présenté plusieurs techniques et quelques théories de la cryptographie qui vont nous permettre de comprendre cet axe de recherche. Nous avons défini la cryptographie, son utilisation et leur objectif. Nous avons abordé les différents types de la cryptographie.

Nous avons parlé sur la cryptographie des images satellitaires notamment : définition et les différents types d'images satellitaires.

Dans les prochains chapitres, nous étudions sur la programmation parallèle pour faciliter l'application des méthodes de la cryptographie qui peut prendre beaucoup de temps. Ainsi nous suggérons une méthode de chiffrement/déchiffrement pour ce type d'image.

Chapitre II : Le parallélisme

II.1 Introduction

Les systèmes parallèles sont devenus la norme, qu'il s'agisse des systèmes de calcul ultra-hautes performances, des ordinateurs personnels ou des serveurs web **[20]**.

Ce chapitre aborde une définition du parallélisme avec ses objectifs principaux, ainsi que les modèles et les architectures parallèles.

II.2 Terminologie

Tâche : une portion de travail à exécuter sur un ordinateur, du type un ensemble d'instructions d'un programme qui est exécuté sur un processeur.

Tâche parallèle : une tâche qui peut s'exécuter sur plusieurs processeurs, sans risque sur la validité des résultats.

Exécution séquentielle : exécution d'un programme séquentiel, une étape à la fois.

Exécution parallèle : exécution d'un programme par plusieurs tâches, chaque tâche pouvant exécuter la même instruction ou une instruction différente, à un instant donné.

Mémoire partagée : D'un point de vue hard, réfère à une machine dont tous les processeurs. Ont un accès direct à une mémoire commune (généralement via un bus). D'un point de vue modèle de programmation : toutes les tâches ont la même image mémoire et peuvent directement adresser et accéder au même emplacement mémoire logique, peu importe où il se trouve en mémoire physique.

Mémoire distribuée : d'un point de vue physique, basée sur un accès mémoire réseau pour une mémoire physique non commune. D'un point de vue modèle de programmation, les tâches ne peuvent voir que la mémoire de la machine locale et doivent effectuer des communications pour accéder à la mémoire d'une machine distante, sur laquelle d'autres tâches s'exécutent.

Communications : les tâches parallèles échangent des données, par différents moyens physiques : via un bus à mémoire partagée, via un réseau. Quelque soit la méthode employée, on parle de « communication ».

Synchronisation : la coordination des tâches en temps réel est souvent associée aux communications, elle est souvent implémentée en introduisant un point de synchronisation au-delà duquel la tâche ne peut continuer tant qu'une ou plusieurs autres tâches ne l'ont pas

modèles en comportent plusieurs. Il y a même des ordinateurs avec des milliers de processeurs. Avec les ordinateurs monoprocesseurs, il est possible d'effectuer un traitement parallèle en connectant les ordinateurs d'un réseau. Cependant, ce type de traitement en parallèle nécessite un logiciel très sophistiqué appelé traitement distribué **[22].**

II.3.1 Objectif de la programmation parallèle

- Gagner du temps et / ou de l'argent: En théorie, le fait de consacrer plus de ressources à une tâche raccourcit son temps de réalisation, avec des économies potentielles ;
- Permet la résolution de problèmes plus importants : De nombreux problèmes sont si grands et / ou complexes qu'il n'est pas pratique ou impossible de les résoudre sur un seul ordinateur, surtout compte tenu de la mémoire de l'ordinateur est limitée ;
- Fournir une concurrence: Une seule ressource de calcul ne peut faire qu'une chose à la fois. Les ressources multiples de calcul peuvent faire beaucoup de choses simultanément **[23]**.

II.3.2 Type de parallélismes

Le parallélisme de données, réalisé par les architectures SIMD, consiste à faire appliquer par un ensemble de tâches la même série d'instructions sur des données différentes. Sur architecture MIMD, on peut distinguer deux grands types de parallélisme ***de tâches*** :

II.3.2.1 Parallélisme Synchrone :

C'est un modèle qui a été utilisé dans nos travaux sur architecture SMP à mémoire partagée, toutes les tâches sont globalement séquentialisées: les lancements, synchronisations et fusions sont communes.

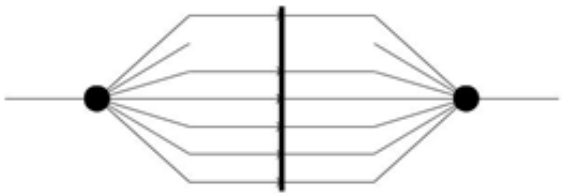


Fig. II.01. Parallélisme synchrone.

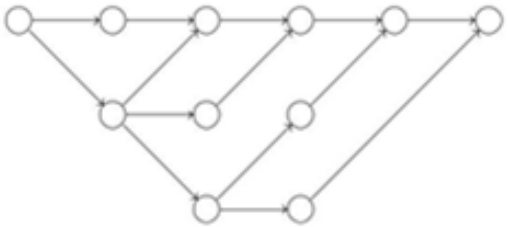


Fig. II.02. Parallélisme asynchrone.

II.3.3 Architecture des parallélismes

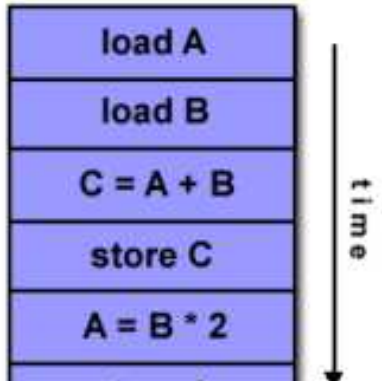
II.3.3.1 Classification de Flynn's

On distingue classiquement quatre types principaux de parallélisme (Taxonomie de Flynn-Tanenbaum): SISD, SIMD, MISD et MIMD. Cette classification est basée sur les notions de flot de contrôle (deux premières lettres, I voulant dire ``Instruction'') et flot de données (deux dernières lettres, D voulant dire ``Data'') **[25]**.

A. Machine SISD

Une machine SISD (Single Instruction Single Data) est ce que l'on appelle d'habitude une machine séquentielle, ou machine de Von Neuman. Une seule instruction est exécutée à un moment donné et une seule donnée (simple, non-structurée) est traitée à un moment donné **[25]**.

Exemples: anciens cadres généraux de génération, postes de travail, PC



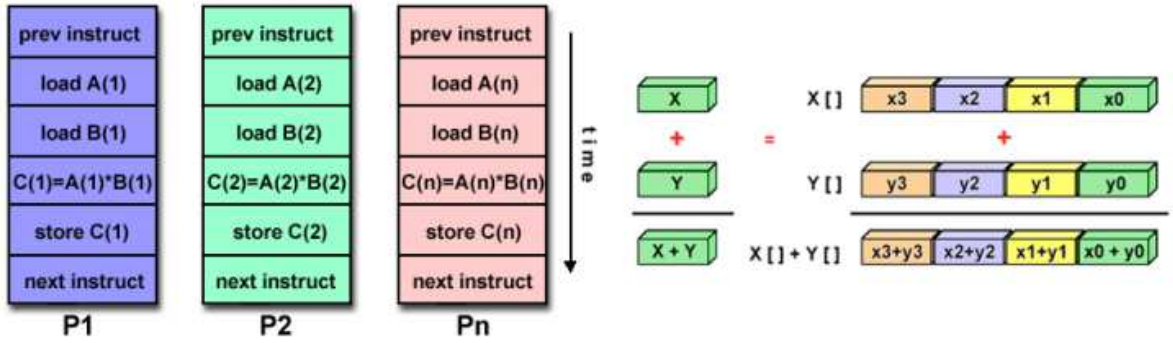


Fig. II.04. Machine SIMD **[26]**.

C. Machine MISD

La machine MISD « Multiple Instruction Single Data » à un seul flux de données est alimenté en plusieurs unités de traitement, chaque unité de traitement opère sur les données indépendamment via des instructions.

Exemple : carnegie-mellon C.mmpComputer (1971)

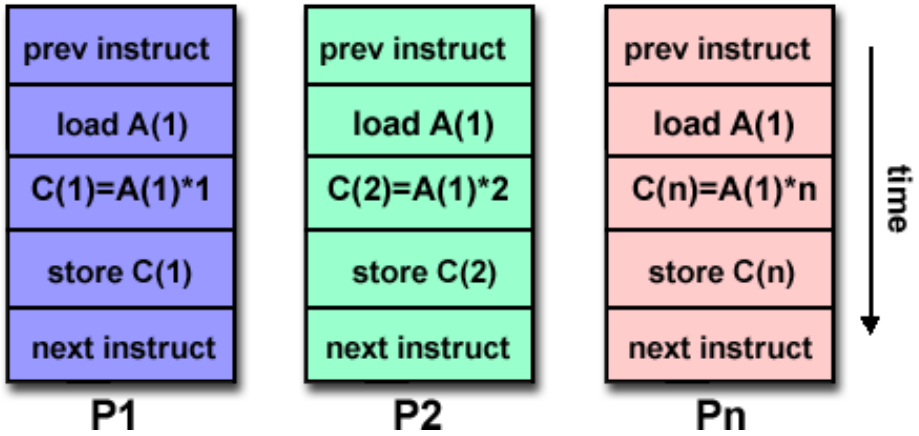


Fig. II.05. : Machine MISD **[26]**.

D. Machine MIMD

Actuellement, la machine MIMD « Multiple Instruction Multiple Data » est le type le plus

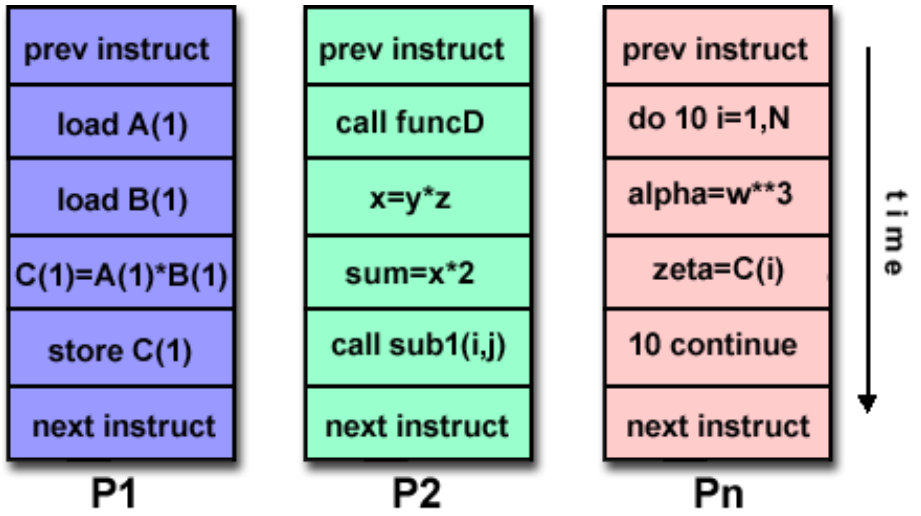
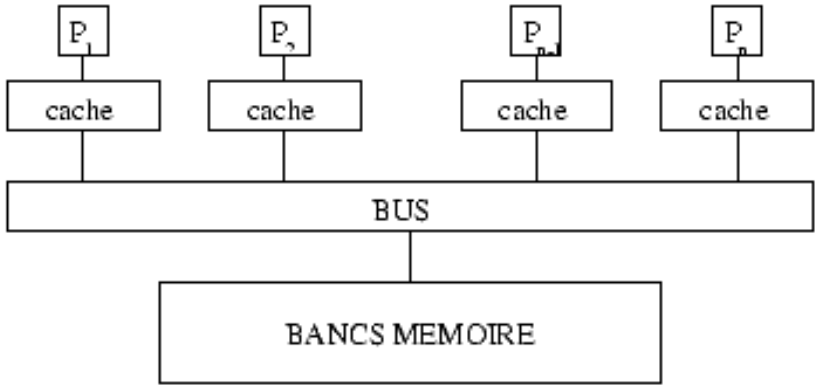


Fig. II.06. : Machine MIMD [26].

Note : beaucoup d'architecture MIMD inclut l'exécution SIMD sous-composantes **[26]**.

II.3.3.2 Architectures à mémoire partagée(SMP)

Une architecture à mémoire partagée est principalement constituée de processeurs avec des horloges indépendantes, donc évoluant de façon asynchrone, et communicante en écrivant et lisant des valeurs dans une seule et même mémoire (la mémoire partagée) **[25]**.



Exemple : processeurs dual core

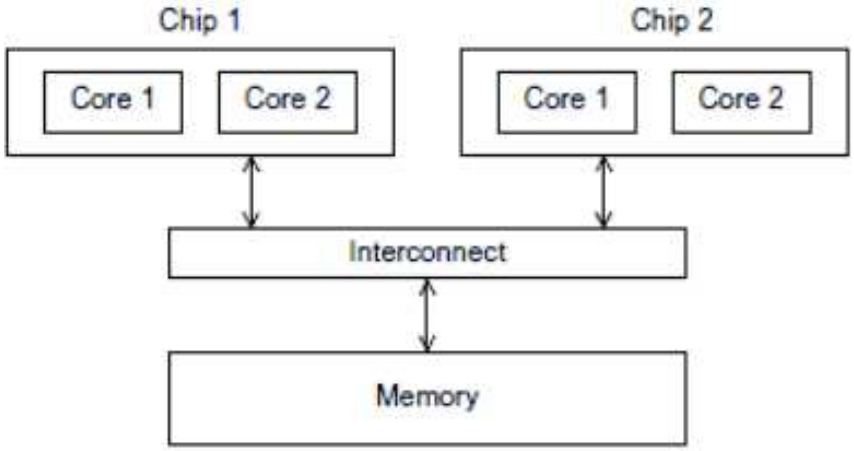
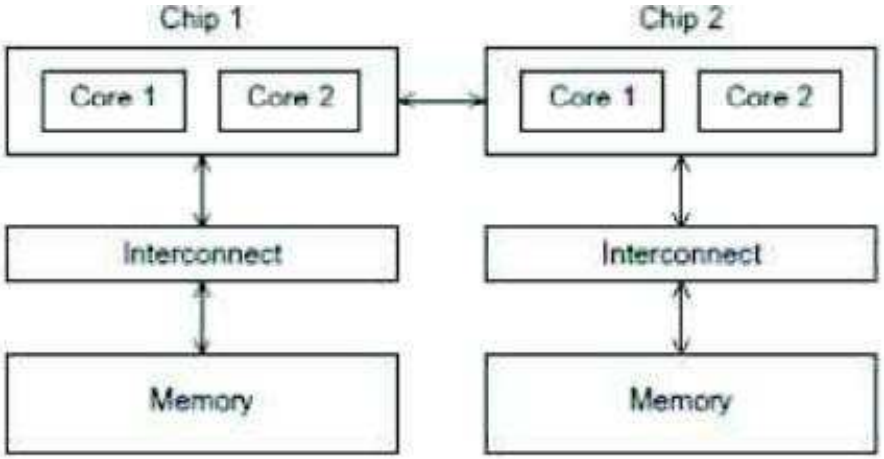


Fig. II.08. Architecture à mémoire partagée UMA **[27]**.

B. NUMA

- La mémoire globale est physiquement distribuée mais logiquement partagée ;
- Un cœur accède plus rapidement à sa mémoire locale qu'au block de mémoire d'un autre cœur.

Exemple : AMD Athlon 64, AMD Opteron, SGI Origin 3000



- Programmeur responsable pour assurer un accès "correct" de la mémoire globale **[26]**.

II.3.3.3 Architectures à mémoire distribuée

Les systèmes de mémoire distribuée nécessitent un réseau de connecter la mémoire entre processeurs.

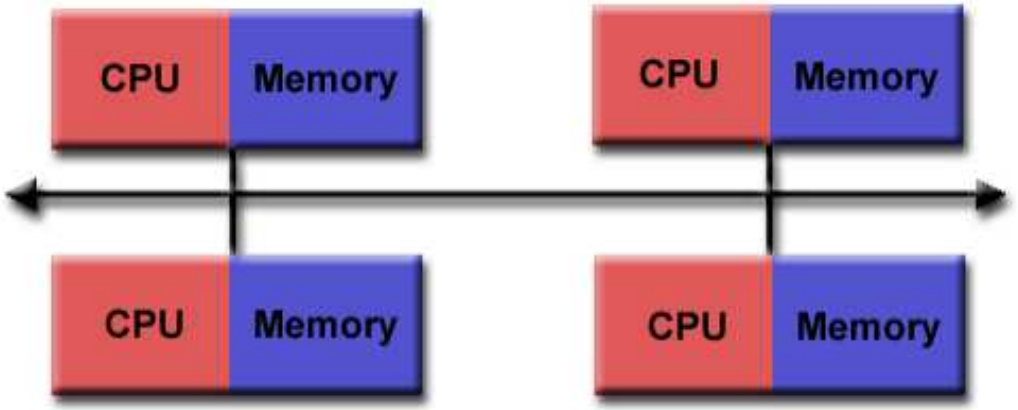


Fig. II.10. Architecture à mémoire distribuée **[26]**.

A. Avantage d'architecture à mémoire distribuée :

- La mémoire est évolutive avec le nombre de processeurs ;
- Aucune interférence de mémoire ou surcharge pour essayer de conserver la cohérence du cache ;
- Coût efficace.

B. Inconvénient d'architecture à mémoire distribuée :

- Programmeur responsable de la communication de données entre les processeurs **[26]**.

II.3.3.4 Architectures mixtes

Aujourd'hui, la plupart des calculateurs combinent les architectures à mémoire partagée et à mémoire distribuée. Ces machines sont alors constituées de machines à mémoire partagée (nœud de calcul) reliées entre elles par un réseau d'interconnexion :

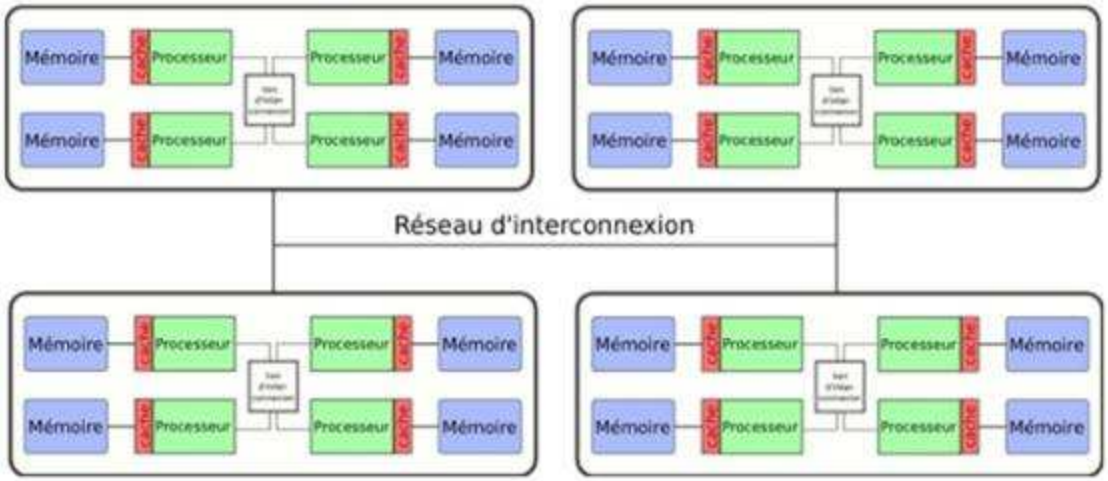


Fig. II.11. Architectures mixte **[28]**.

II.3.3.5 Architectures hybrides

Les ordinateurs les plus puissants au monde sont aujourd'hui un mixte de mémoire partagée et mémoire distribuée.

La brique de base (nœud) est un multiprocesseur à la brique de base (nœud) est un multiprocesseur à mémoire partagée. Ces briques sont interconnectées par un réseau (type Ethernet, Myrinet, Infiniband,) **[29]**.

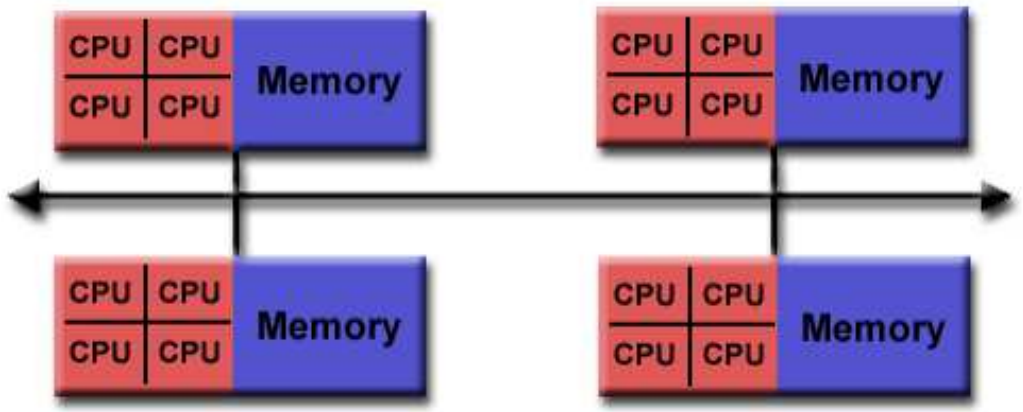


Fig. II.12. Architecture hybride **[26]**.

II.3.4.2 Architectures à mémoire distribuée :

Sockets bibliothèque standardisée, bas niveau ;
MPI Message Passing Interface, bibliothèque standard de fait pour les architectures à mémoire distribuée (fonctionne aussi sur les SMP), remaniement important du code **[30]**.

II.4 Conclusion

Dans ce chapitre nous avons vu la définition et l'objectif de la programmation parallèle, aussi quelque terme de parallélisme.

Nous avons évoqué les modules et l'architecture des ordinateurs qui permettent de faire le parallélisme des programmes.

Dans le chapitre suivant, nous allons éclairer les méthodes utilisées dans notre projet d'une façon détaillée.

Chapitre III : Les méthodes utilisées

III.1 Introduction

Les travaux de recherche de ce chapitre s'inscrivent dans le cadre de crypter des images satellitaires, par l'algorithme de RSA. Cet algorithme est fondé sur l'utilisation d'une paire de clés composée d'une clé publique pour chiffrer et d'une clé privée pour déchiffrer des données confidentielles, et la mémoire partagée ça sera notre architecture pour faire le parallélisme, en effet qu'il y a d'autre architecture à citer comme l'architecture à mémoire distribuée, mixte et hybride.

III.2 Méthode de cryptage

Nous allons développer un petit programme de cryptage basé sur ce chiffre. Le but ne sera pas de développer un programme au code « incassable », mais plutôt de comprendre comment fonctionne le cryptage RSA.

III.2.1 Vue historique

Le système de cryptage RSA a été inventé en 1977 par Ron Rivest, Adi Shamir et Len Adleman (dont les initiales forment RSA). Ces 3 auteurs avaient décidé de travailler ensemble pour établir qu'un nouveau système de codage révolutionnaire, dénommé «système à clef publique» que W.Diffie et M.hellman venaient d'inventer, était une impossibilité logique (autrement dit, que tout système de cryptage de cette nature présentait des failles). Ils ne réussirent pas dans leur projet, mais, au contraire, découvrirent un nouveau système à clef publique qui supplanta vite celui de W.Diffie et M.Hellman **[31]**.

III.2.2 Cryptosystème à clé publique

Les algorithmes asymétriques sont basés sur une clef pour le chiffrement et une clef associée différente, pour le déchiffrement. Ces algorithmes ont la caractéristique importante suivante. Il est impossible de trouver la clef de déchiffrement malgré la connaissance de l'algorithme cryptographique et la clef de chiffrement.

III.2.3 L'algorithme RSA

Le chiffrement RSA est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet **[32]**.

III.2.4.1 Génération des clés

Cette phase peut se résumer en 03 étapes:

1ére étape : Alice choisit au hasard deux grands nombres entiers, naturels, premiers, p et q. Dans notre exemple simplifié elle choisit:

p = 31 et q = 53

Et fait leur produit:

n =p*q = 1643

2ème étape: Alice détermine la fonction d'Euler associée à n déjà calculé en utilisant la formule:

$\phi(n)= (p-1)*(q-1)$

$\phi(n)=30*52=1560$

Une fois que la fonction d'Euler déterminée, Alice choisie au hasard sa clé publique « e », cette clé est un nombre premier compris entre 1 et $\phi(n)$ et premier relativement à $\phi(n)$ c'est-à-dire le PGCD (e, $\phi(n)$)=1. Alice fait: e=11

D'où le couple (e, n) constitue la clé publique.

$$\begin{cases} \text{n: c'est le module} \\ \text{e: c'est l'exposant} \end{cases}$$

La clé publique est donc (11, 1643).

3ème étape: Cette dernière étape consiste à trouver la clé privée « d » qui correspond à la clé publique choisie précédemment avec d compris entre 1 et $\phi(n)$, pour se faire, il faut résoudre l'équation suivante:

$e*d \bmod \phi(n)=1$

c.à.d: $e*d \equiv 1 \ [\phi(n)]$

Donc: $e*d= K.\phi(n) +1$

{ n: c'est le module
{ d : c'est l'exposant

En fin, Alice et Bob disposent toutes les clés indispensables au chiffrement et au déchiffrement des messages après la transmission ou la publication de sa clé publique (e, n).

Maintenant, il faut qu'elle conserve sa clé privée (d, n) et qu'elle n'oublie jamais les nombres p et q.

III.2.4.2 Chiffrement

Bob veut donc transmettre le message M « ANEMONE » à Alice. Il crypte M à partir de la clé public (dans notre exemple n= 1634 et e=11) qui a généré dans la phase précédant.

Il va procéder au cryptage de la manière suivante :

Il transforme en nombres son message en remplaçant par exemple chaque lettre par son rang dans l'alphabet : a= 01, b= 02…..z= 26, il résulte :

M= ANEMONE

M= A N E M O N E

M= 01 14 05 13 15 14 05

Il découpe son message numérisé en blocs de même longueur représentant chacun une taille égale ou inférieure à celle de n ce qui empêche la simple substitution. Dans notre exemple la taille de n est 3, ce qui donne des tranches m_i de 03 chiffres chacune, le message devient :

M= 001 140 513 151 405

M= m_1 m_2 m_3 m_4 m_5

Chaque bloc m_i est chiffré par l'équation :

$$Ci=m_i^e \bmod n$$

Ce qui donne :

$$C_1 = m_1^{11} \bmod 1643 = 001$$

001^{11} mod 1643=	1
140^{11} mod 1643=	109
513^{11} mod 1643=	890
151^{851} mod 1643=	1453
405^{11} mod 1643=	374

Alors le message chiffré C sera :

C= c_1 c_2 c_3 c_4 c_5

C= 0001 0109 0890 1453 0374

Enfin, Bob a son message chiffré, il peut donc l'envoyer à Alice.

III.2.4.3 Déchiffrement

Alice reçoit le message de Bob, à partir de p et q, qu'elle a gardés secrets elle calcule la clef d de déchiffrement (c'est sa clef privée). Celle-ci doit satisfaire l'équation:

e*d mod ((p-1) (q-1))=1

Chacun des blocs c_i du message chiffré sera déchiffré par l'équation:

$m_i=c_i^d$ mod n

Ce qui lui donne :

1^{851}mod 1643=	1
109^{851}mod 1643=	140

Lors du déchiffrement, sachant qu'il faut obtenir des blocs de 2 éléments (grâce au codage particulier de l'exemple). Finalement, Alice prend sa table de correspondance alphabétique pour restituer le message M, elle aura:

01 14 05 13 15 14 05

A N E M O N E **[32]**.

III.2.5 Avantage de l'algorithme RSA

- RSA est l'un des systèmes de cryptage asymétrique ayant le plus de succès de nos jours ;
- L'élimination de la problématique de la transmission de la clé ;
- La possibilité d'utiliser la signature électronique ;
- L'impossibilité décrypter le message dans le cas de son interception par une personne non autorisée **[33]**.

III.2.6 Inconvénient de l'algorithme RSA

- Lent, comme tout système asymétrique;
- Attaques par substitution de clé possibles **[33]**.

III.3 Méthode de parallélisme

Le parallélisme fait coopérer plusieurs processeurs pour réaliser un calcul, l'exécution d'un algorithme en utilisant plusieurs processeurs, division d'un algorithme en tâches exécutables simultanément. D'après notre recherche on a choisi de travailler avec l'architecture à mémoire partagé « SMP ».

Il y a plusieurs outils qui permettent de faire le parallélisme à mémoire partagée, parmi ses outils il y a la bibliothèque OpenMP.

III.3.1 OpenMP

C'est une interface de programmation pour le calcul parallèle sur une architecture à mémoire

- parallélisation de haut niveau **[35]**.

III.3.3 Inconvénients OpenMP:

- Coût ;
- limitation du nombre de processeurs (conflit d'accès au niveau matériel) ;
- la bande passante du réseau est le facteur limitant de ces architectures **[35]**.

III.4 Conclusion

Dans ce chapitre nous avons parlé de notre choix d'algorithme de cryptage et le type machine parallèle, on a choisis l'algorithme RSA comme une méthode de cryptage et la machine à mémoire partagée pour le parallélisme en se basant sur les recherches que nous avons effectué dans les chapitres précédents.

On va les implémenter dans le chapitre 4.

Chapitre IV : Application et résultat

IV.1. Introduction

Nous avons vu dans les chapitres précédents, les différents systèmes de cryptage, décryptage et leurs fonctionnements.

Ce chapitre porte sur le développement de l'application où on va expliquer le chiffrement ou le déchiffrement d'une image satellitaire en utilisant l'algorithme de chiffrement asymétrique RSA avec la programmation parallèle, en détaillant les différentes étapes par laquelle une image passe avant d'être cryptée, ainsi que l'environnement de programmation, en illustrant des résultats de traitement en se basant sur des analyses et des tests.

IV.2. Environnement de développement

A. Ressources matérielles

- Processeur Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz
- Mémoire installée (RAM) : 4.00Go
- Type de système: Système d'exploitation 64 bits

B. Ressources logicielles

- Système d'exploitation : Windows 7
- Editeur utilisé : Visual Studio Community 2015, QT version 4.1.2

IV.3. Outils de développement

A. Langage C++ :

C++ est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plates-formes matérielles et de systèmes d'exploitation.

Chapitre IV : Application et résultat

B. Microsoft Visual Studio :

C'est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle **Visual Studio 2017**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

Durant sa conférence *Connect*() 2016, Microsoft a annoncé le portage de Visual Studio sur macOS, le système d'exploitation d'Apple [37].

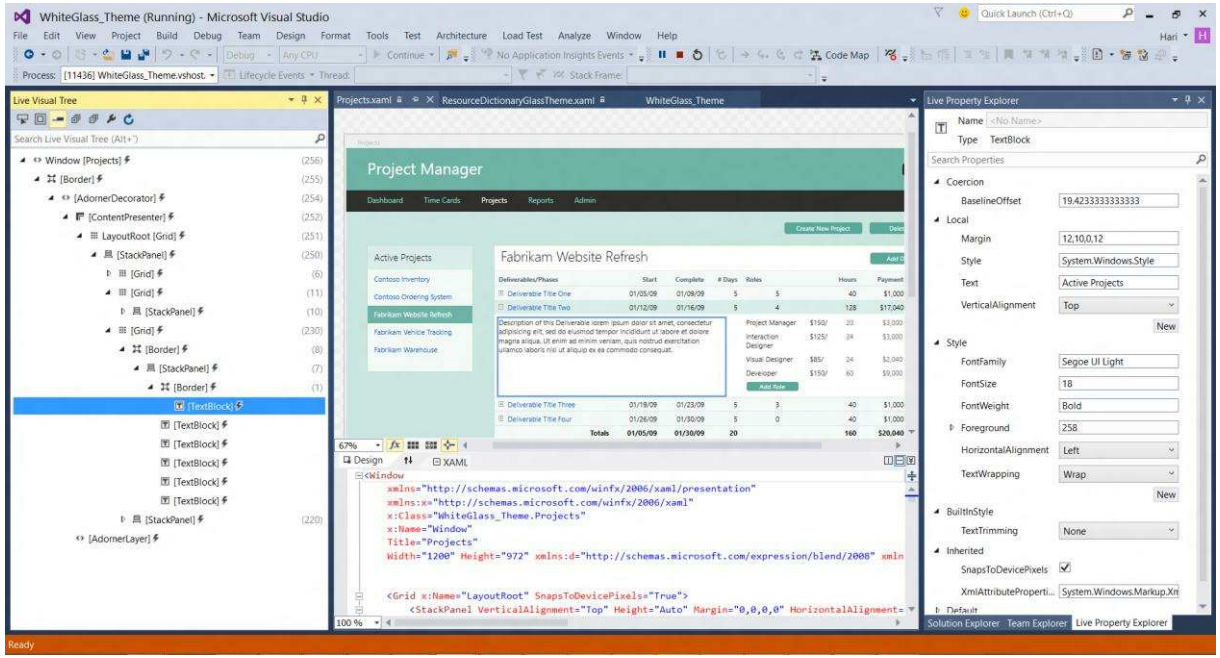


Fig. IV.01 Interface de Visual Studio 2015

Chapitre IV : Application et résultat

Qt permet la portabilité des applications qui n'utilisent que ses composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont GNU/Linux) qui utilisent le système graphique X Window System ou Wayland, Windows, Mac OS X et également Tizen. Le fait d'être une bibliothèque logicielle multiplateforme attire un grand nombre de personnes qui ont donc l'occasion de diffuser leurs programmes sur les principaux OS existants.

Qt supporte des bindings avec plus d'une dizaine de langages autres que le C++, comme Ada, C#, Java, Ruby, Visual Basic, etc.

Qt est notamment connu pour être le framework sur lequel repose l'environnement graphique KDE, l'un des environnements de bureau par défaut de plusieurs distributions GNU/Linux [39].

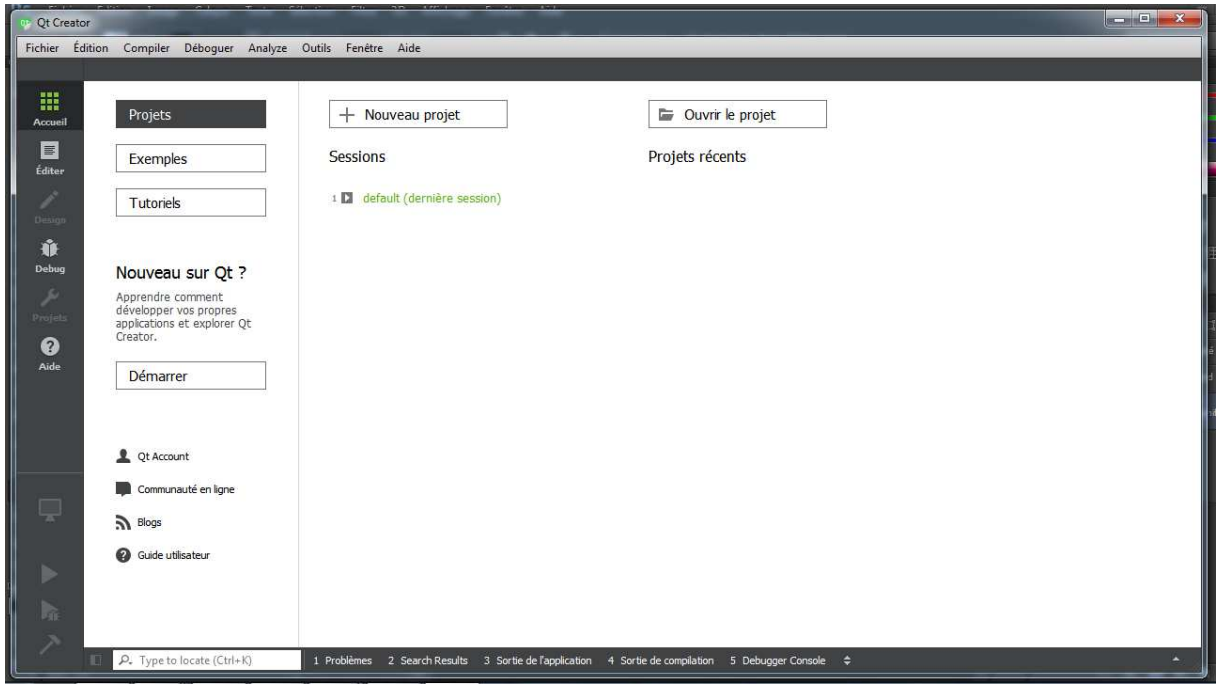
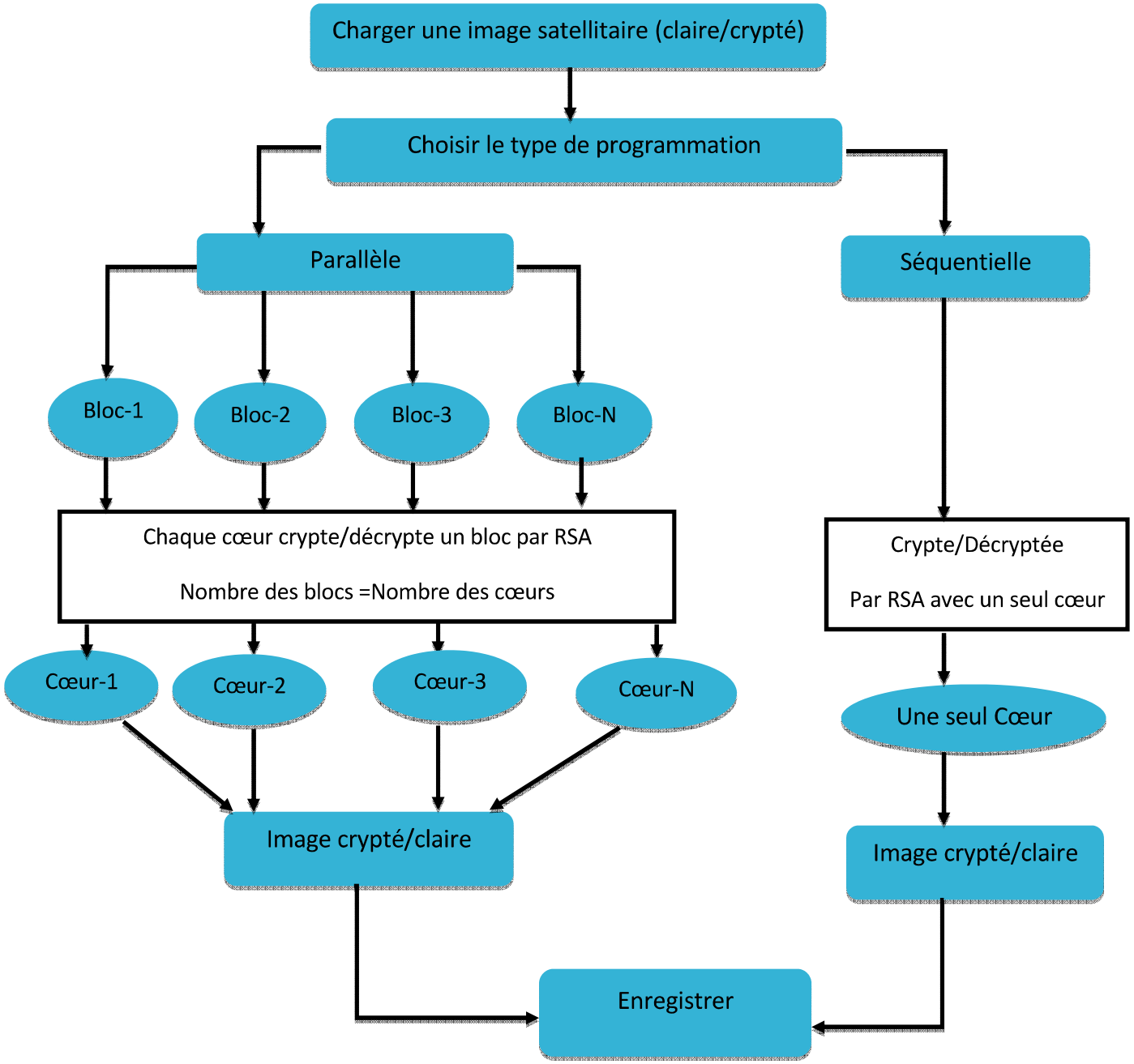


Fig. IV.02. interface QT.

IV.4 Description de l'application

Chapitre IV : Application et résultat



Chapitre IV : Application et résultat

Si on choisi de crypter en appuyant sur le bouton « cryptage » sinon en appuyant sur la bouton « décryptage».

Fig. IV.04. Fenêtre principale

IV.5.1.Cryptage

Pour le chiffrement en appuyant sur le bouton « cryptage », on obtient l'interface suivante :

Chapitre IV : Application et résultat

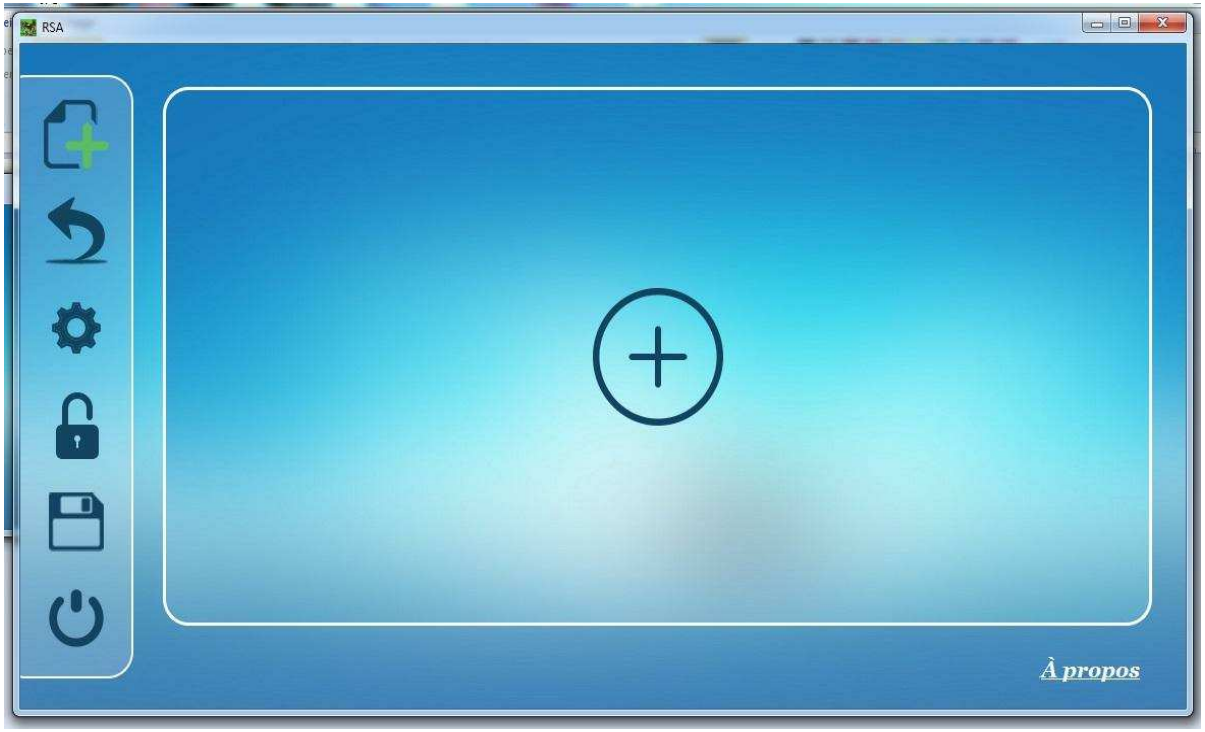
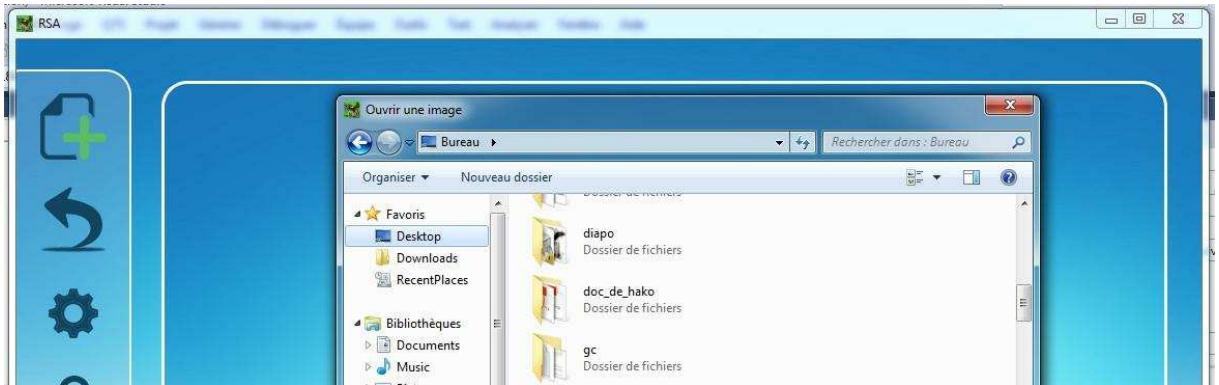


Fig. IV.05. Fenêtre principale de cryptage

Tout d'abord il faut ouvrir une image en cliquant sur le bouton « ouvrir »



Chapitre IV : Application et résultat

On va sélectionner une image satellitaire à partir de google earth de l'année (2017) de système de référence WGS 1984-UTM zone 31.

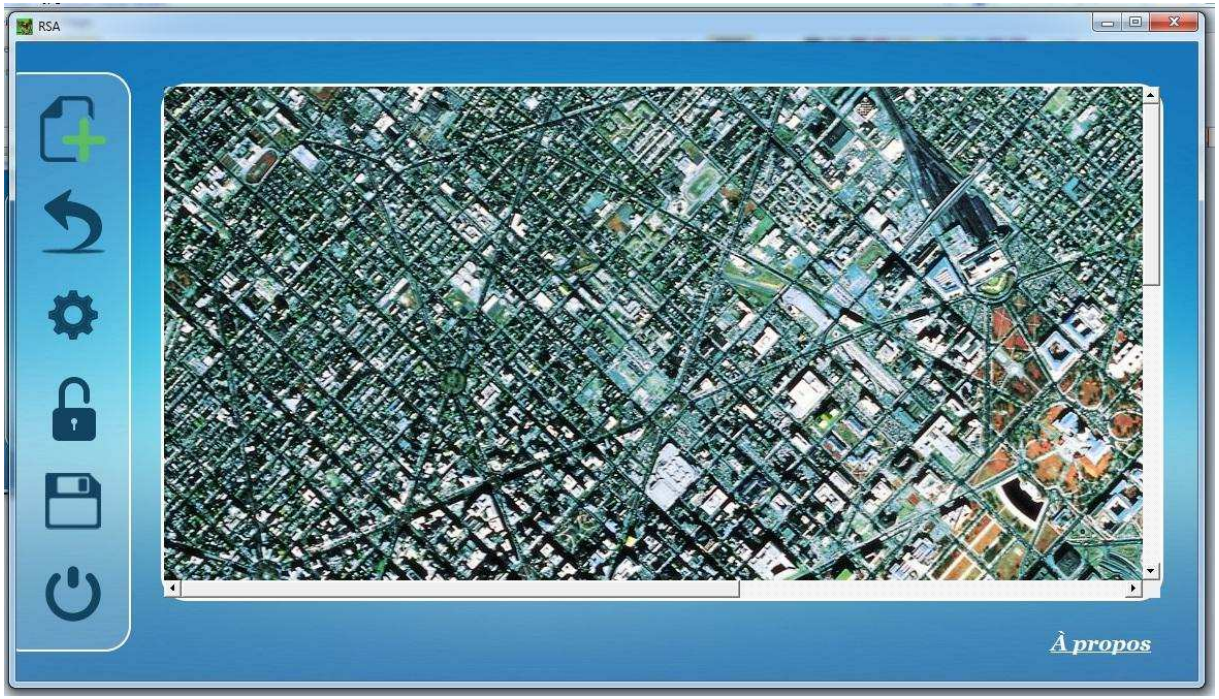
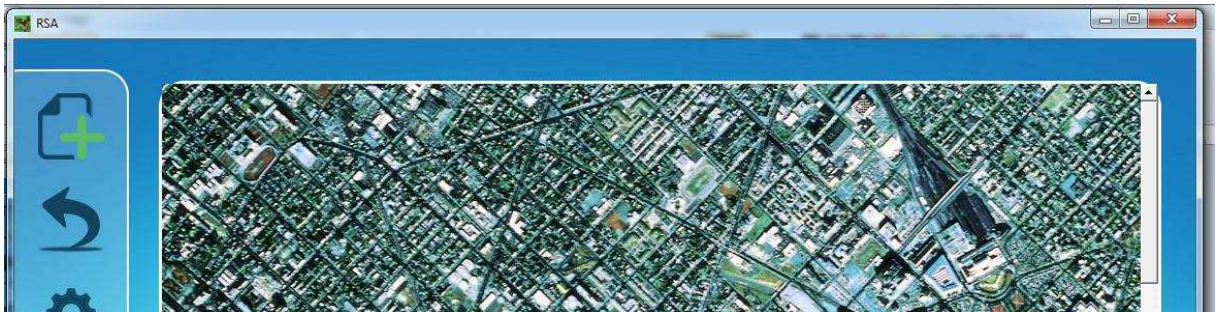


Fig. IV.07. Sélectionner image satellitaire

Après avoir chargé l'image on cliquant sur le bouton « crypté » là ou li est placé le curseur.



Chapitre IV : Application et résultat

On peut afficher les paramètres de cryptage en cliquant sur le bouton  cette paramètre contient deux type de programmation (séquentiel/parallèle), le type de programmation est automatiquement en parallèle.

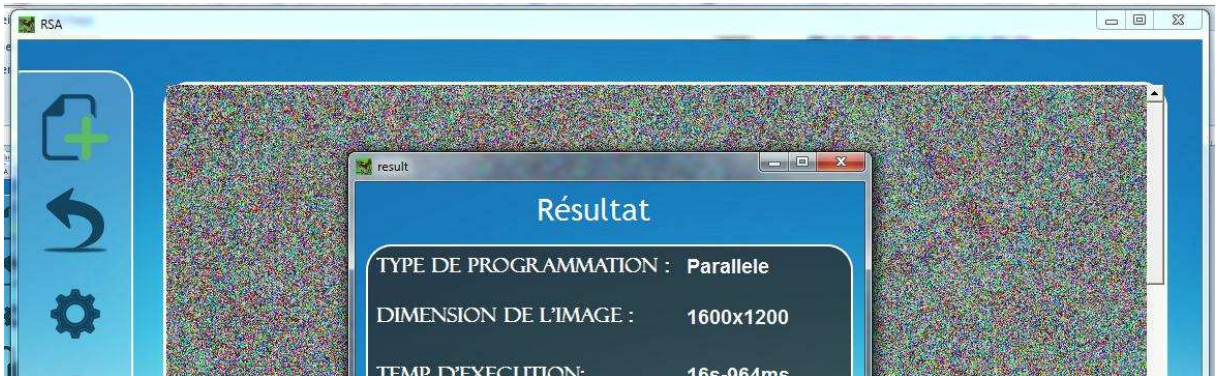
Fig. IV.08. Paramètres de cryptage

Une fois cliqué sur le bouton de cryptage une fenêtre de résultat sera affichée.

Chapitre IV : Application et résultat

Si on veut le crypter dans le mode séquentiel on doit modifier les paramètres de cryptage, choisir « séquentiel ».

Fig. IV.10. Paramètres de cryptage séquentiel

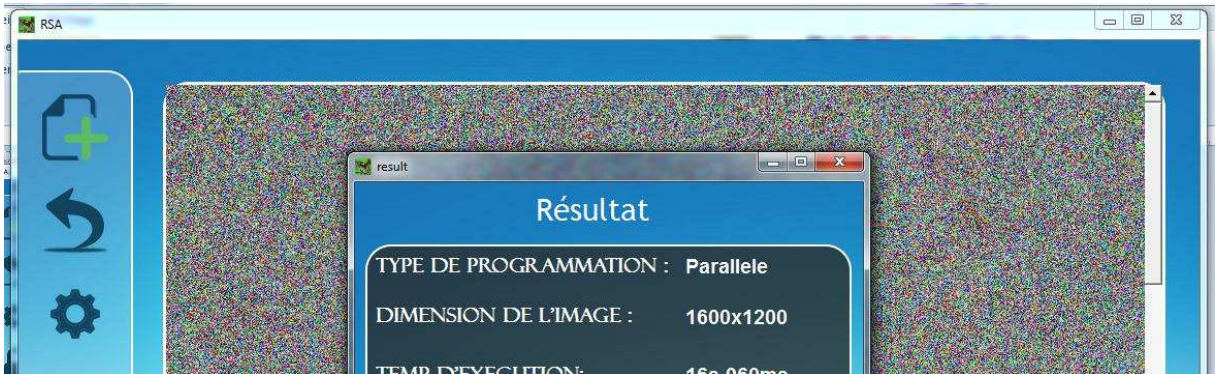


Chapitre IV : Application et résultat

L'algorithme RSA ne crypte pas tous les pixels donc si on veut crypter tous les pixels on coche sur « crypter tous les pixels » dans le paramètre de cryptage.

Fig. IV.12. Paramètres de crypter tous les pixels

Une fois lancer le cryptage Le taux sera 100 %.



Finalement, on enregistre l'image crypté on cliquant sur le bouton « enregistrer ».

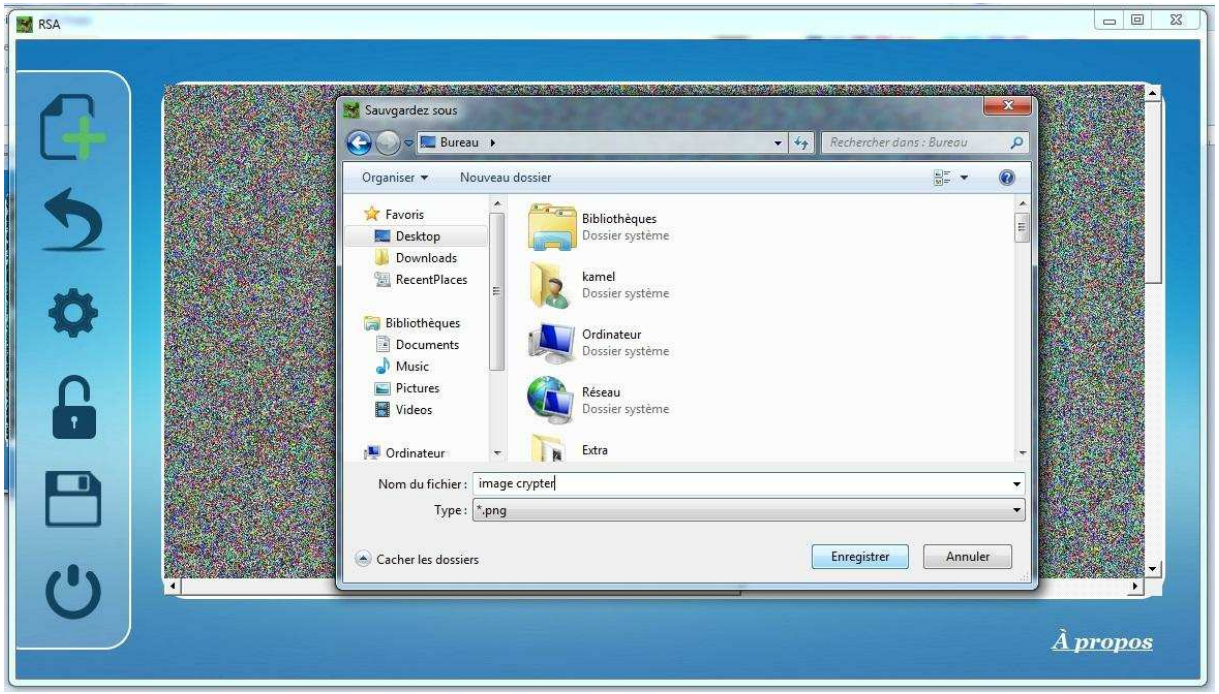
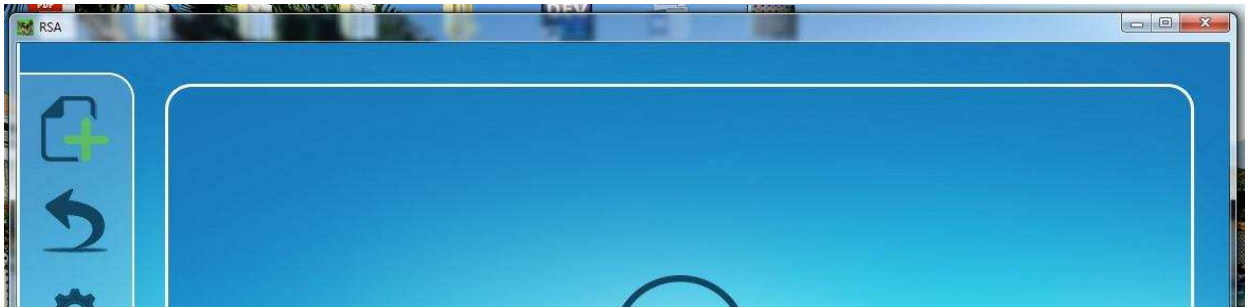


Fig. IV.14. Enregistrer l'image cryptée.

IV.5.2. Décryptage

Si on choisi le décryptage en cliquant sur le bouton « décrypté »,on obtient l'interface suivant :



Chapitre IV : Application et résultat

Il faut ouvrir une image en cliquant sur le bouton « ouvrir » :

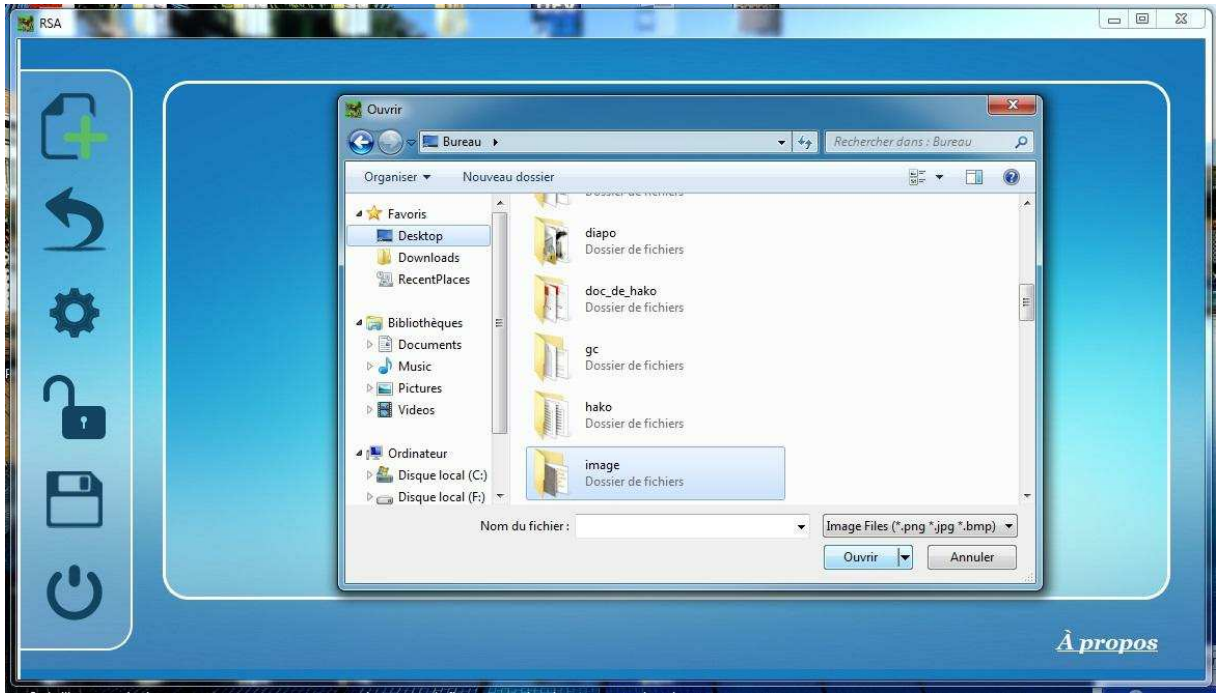
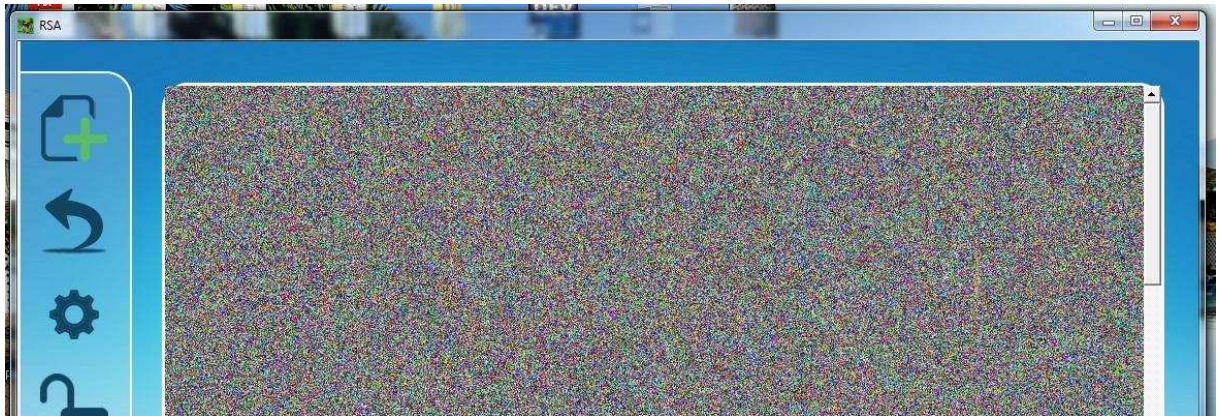


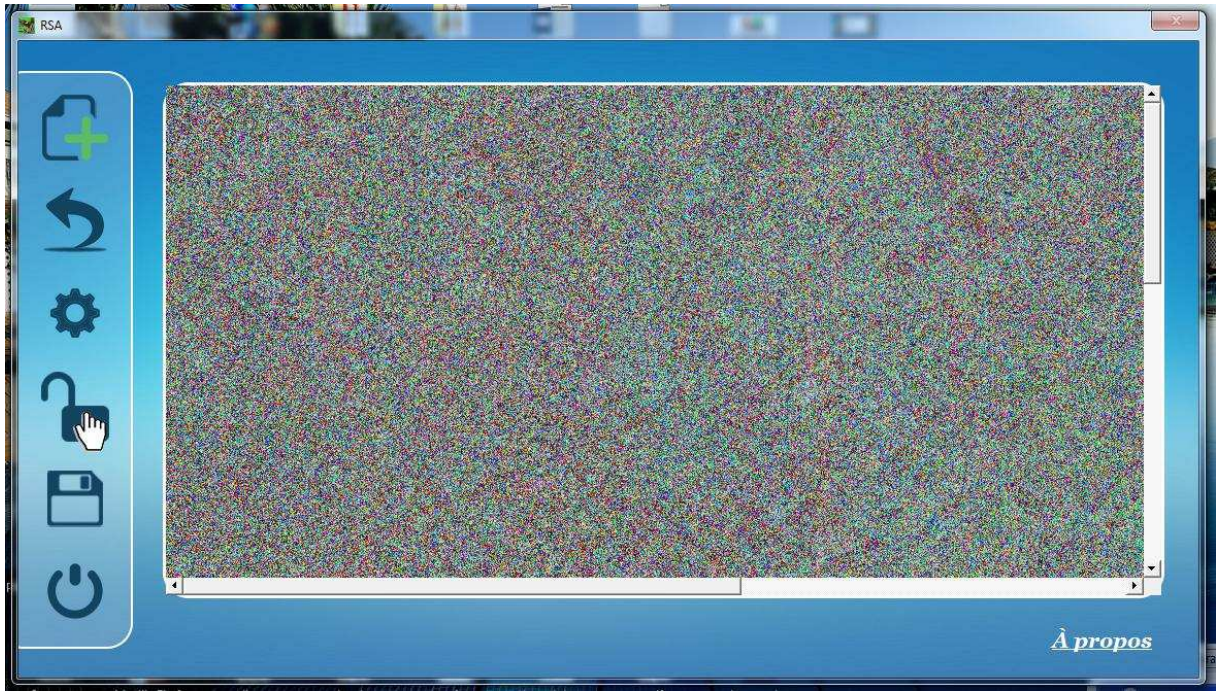
Fig. IV.16. Ouvrir une image

On va sélectionner une image déjà cryptée.



Chapitre IV : Application et résultat

Après avoir chargé l'image on cliquant sur le bouton « décrypté » là ou li est placé le curseur.



On peut afficher les paramètres de décryptage en cliquant sur le bouton  cette paramètre contient deux type de programmation (séquentiel/parallèle), le type de programmation est automatiquement en parallèle.

Chapitre IV : Application et résultat

Une fois cliqué sur le bouton de décryptage une fenêtre de résultat sera affichée.

Fig. IV.19. Résultat de décryptage avec parallélisme.

Si on veut décrypter avec la programmation séquentielle on doit cocher sur « séquentielle ».

Chapitre IV : Application et résultat

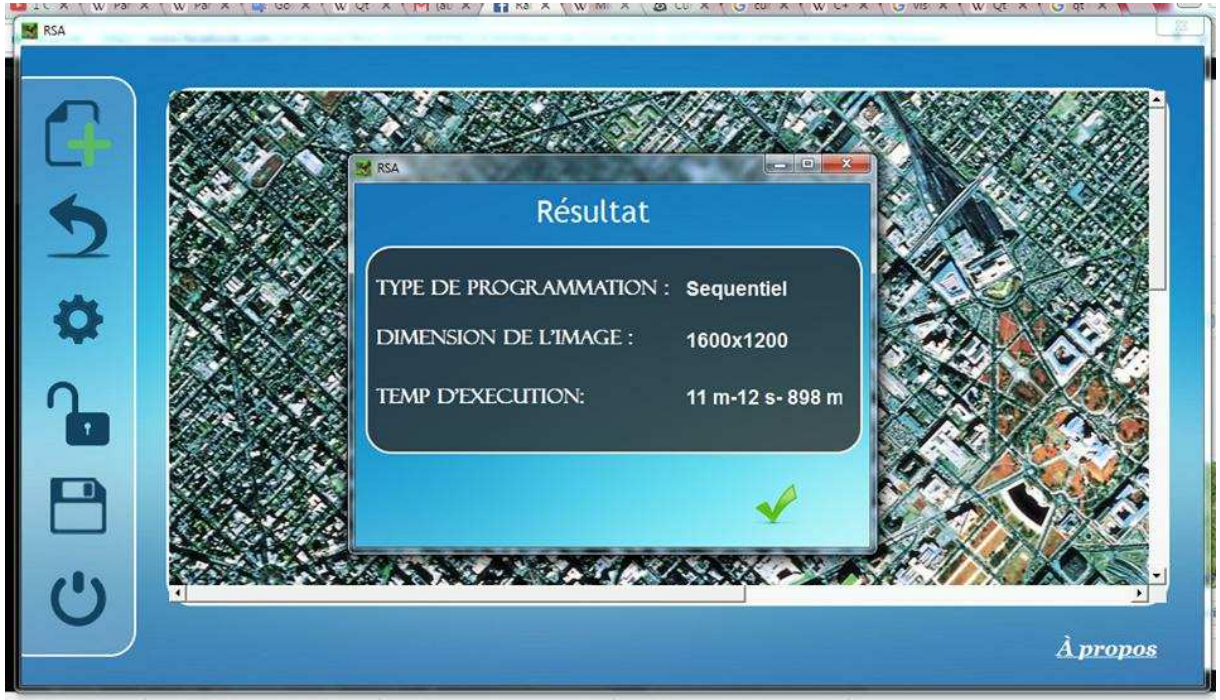
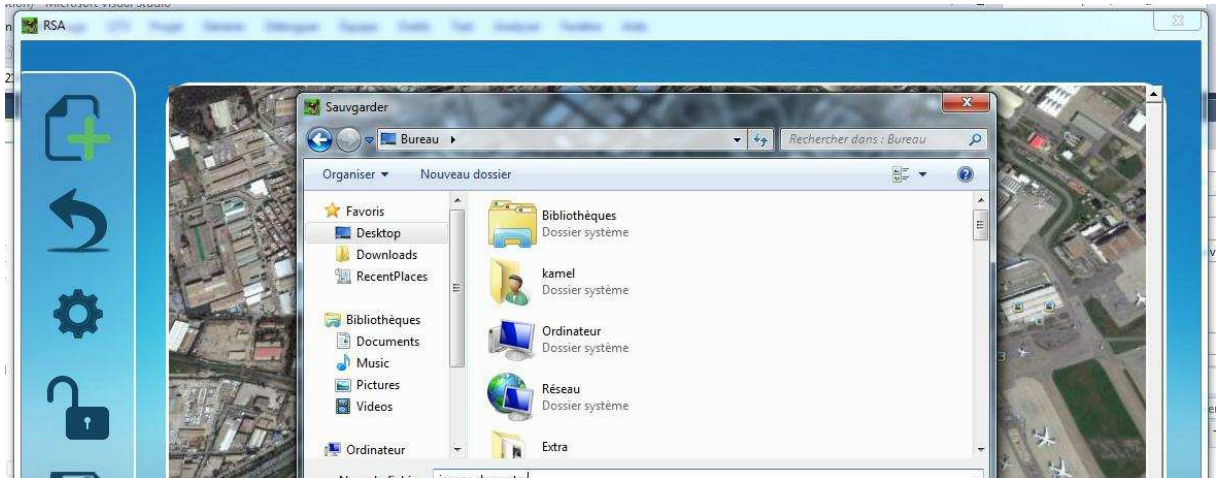


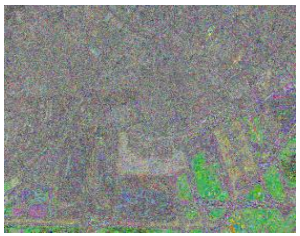
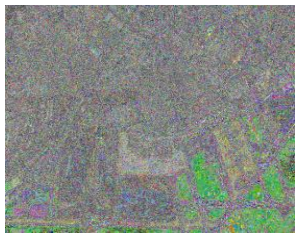
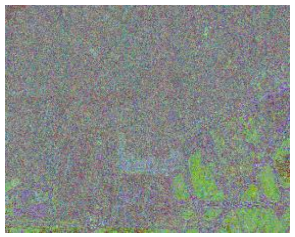
Fig. IV.21. Résultat de décryptage séquentiel.

Si on veut enregistrer l'image décryptée en cliquant sur le bouton « enregistrer»

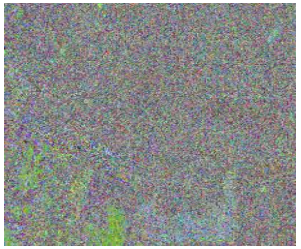
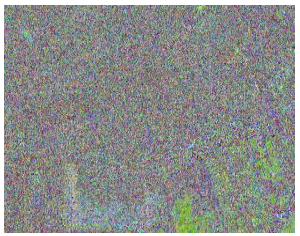


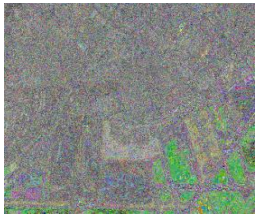
Chapitre IV : Application et résultat

IV.6. Résultats de chiffrement/déchiffrement :

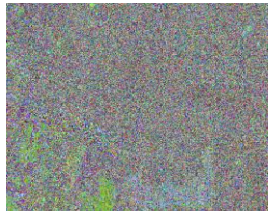
Cryptage					
Image originale	**Image cryptée**	**Type**	**Crypter Tous les pixels**	**Taux %**	**Temps**
Image de dimension = 1047 x 561					
		**séquentiel**	**Non**	**86.6039**	**13s-977ms**
		**parallèle**	**Non**	**86.6039**	**5s-222ms**
		**parallèle**	**Oui**	**100**	**5s-222ms**
Image de dimension = 698 x 353					

Chapitre IV : Application et résultat

		**parallèle**	**Non**	**85.1843**	**2s-278ms**
		**parallèle**	**Oui**	**100**	**2s-136ms**

Décryptage				
Image cryptée	**Image décryptée**	**Type**	**Dimension**	**Temps**
Image 1				
		**séquentiel**	**1047 x 561**	**3m-18s-592ms**
		**parallèle**	**1047 x 561**	**1m-8s-725ms**

Chapitre IV : Application et résultat

		**parallèle**	**698 x 353**	**2s-278ms**

❖ **Discussion**

D'après le résultat obtenu pour l'application de notre cryptosystème asymétrique que le cryptage par le parallélisme chiffre plusieurs pixels simultanément selon le nombre des cœurs. Contrairement, le cryptage séquentiel chiffre un seul pixel et utilise un seul cœur.

Si on fait une comparaison entre le cryptage d'une image avec la programmation parallèle et séquentielle on remarque que le temps d'exécution en parallélisme plus rapide que l'exécution séquentielle. Théoriquement :

$$\text{Temps d'execution parallèle} \approx \frac{\text{Temps d'execution sequentiel}}{4}$$

Le taux de cryptage par l'algorithme RSA n'arrive pas à 100%, et pour cela on ajoute une option pour le avoir un taux=100%, mais cette option fait un changement sur quelles que valeurs de l'image, l'utilisation de cette option pour la confidentialité des images est très sensible.

Mais ce qui est le plus important est le fait de ne pas avoir une perte d'information, donc nous somme aussi satisfait pour les résultats obtenus.

IV.7. Conclusion

On peut dire que le mode proposé permet de retourner des bons résultats par rapport à la qualité d'image et le temps d'exécution, aussi crypter et décrypter avec zéro perte de données.

Après le test et les analyses effectués, on a trouvé que cette méthode de chiffrement est solide

Conclusion général

La cryptographie a défini les notions de sécurité et prouvé la sécurité de cryptosystème de chiffrement, de code d'authentification de message et de signature numérique. De plus, des protocoles de pus haut niveau, comme des systèmes de communication sécurisés ou de votes électroniques ont été conçus.

Les progrès réalisés en architecture des processeurs ont permis aux ordinateurs séquentiels de voir leur puissance de calcul doubler environ tous les deux ans, le parallélisme offre une alternative permettant d'augmenter encore d'un facteur de puissance de calcul. Le principe de base de parallélisme est simple. Il consiste à faire travailler ensemble plusieurs processeurs sur un même problème afin de le résoudre plus rapidement.

En général, on peut dire que la méthode proposée permet d'atteindre de très bon résultat de point de vue la qualité d'image et le temps d'exécution. En effet, elle permet de crypter énormément la quantité des informations contenues dans les images satellitaires tout en conservant leurs propriétés originales. Dans plusieurs applications nous n'avons pas intérêt à crypté l'mage dans un peu temps et aussi de crypté l'image avec tous les pixels, mais seulement de chiffrer une image avec un algorithme spécifique. Pour cela on a essayé d'adapter notre application pour réaliser cet objectif et l'ensemble de tests effectués montre la bonne robustesse de différentes combinaisons de l'algorithme de chiffrement aux différents types d'attaques.

Puisque les algorithmes de chiffrement prennent un temps d'exécution lent, donc le parallélisme est devenu comme une technique pour diminuer ce temps.

Comme perspectives, nous proposons que notre cryptosystème de chiffrement basé sur l'algorithme RSA avec le parallélisme soit une application web pour sécuriser la transmission des données cryptées via un réseau.