وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة عبدالحميد ابن باديس مستغانم

Université Abdelhamid Ibn Badis de Mostaganem

كلية العلوم و التكنولوجيا

Faculté des Sciences et de la Technologie

N°d'ordre : M...../GE/2020

MEMOIRE DE FIN D'ETUDES DE

MASTER ACADEMIQUE

FILIERE : TELECOMMUNICATIONS

Spécialité : Système des Télécommunication

Thème

Réalisation d'un système sécurisé de communications numériques Sans fils autour de Raspberry Pi.

Présenté par :

ALLALI Amina

KEDDAR Noura

Soutenu le / 09/ 2020 devant le jury composé de :

Président : Mr.

Examinateur : Mr.

Encadreur : Mr. Mansour Abed

Co-encadreur : Mr.

Année Universitaire : 2019/ 2020

Remerciements

En premier lieu, nous tenons à remercier Allah le tout puissant et le tout miséricordieux de nous avoir donné la santé.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes auxquelles nous tenons à exprimer notre gratitude.

Je voudrais tout d'abord adresser toute ma gratitude à l'encadreur de ce mémoire, Mr. Mansour Abed, pour sa patience, sa disponibilité et surtout ses conseils judicieux, qui ont contribué à alimenter Notre réflexion.

Nous tenons à exprimer notre gratitude à nos parents et frères, qui nous ont apporté un soutien moral et matériel et des discussions animées tout au long de notre parcours.

Je tiens à exprimer ma gratitude à Saad Mohamed Abdeljalil pour sa confiance, sa motivation et son soutien inestimables. De plus, grâce à lui, il a été possible de se détendre pendant cette période mouvementée et le résultat en est la meilleure preuve.

Enfin, nous espérons que ce mémoire sera un encouragement pour les étudiants de l'année prochaine, j'espère que vous apprécierez de le lire. Volonté pour réussir dans nos études.

Dédicaces

Je dédie ce travail à:
À mes parents et à mes frères Amr et Ismail, qui ont été mon plus grand
Soutien.
À tous ceux qui me tiennent à cœur, en particulier à mon père,
Pour son aide, son temps, ses encouragements,
Son aide et son soutien.

ALLALI Amina

Je dédie ce travail à ma mère et mon père qui a été une source de soutien
Et de réconfort pour moi.
À mes frères qui ont contribué à Mon
Encouragement moralement.

KEDDAR Noura

Résumé

Ce mémoire de master traite la réalisation d'un system de communication numérique sans fil bien sécurisé basé sur des mini-ordinateurs de type Raspberry. La sécurité de la transmission est assurée par trois modes à savoir le chiffrement par décalage, le chiffrement par bloc, ainsi que l'étalement de spectre. L'objectif principal est la construction d'un réseau local sans fils, full duplex, à utilisateurs multiples, assurant un transfert sécurisé de données et qu'il soit totalement indépendant de l'internet. La solution proposée pour réaliser ce system est l'utilisation de Raspberry, le cœur du réseau réalisé. Nous avons d'abord configuré un réseau WiFi dans Raspberry Pi 1 (point d'accès), en choisissant le protocole WPA2. En suite, nous nous sommes servis d'un Raspberry Pi 4. Ce dernier permet de gérer les fichiers et de créer le serveur NAS pour partager les données avec d'autres appareils (Smartphones, tablettes, ordinateurs…). La tâche primordiale dans ce projet est bien évidemment la sécurisation des données transmises en implémentant des algorithmes relatifs aux trois modes cités plus avant sous Python et Matlab. Afin de faciliter la tâche à l'utilisateur, nous avons organisé notre projet de fin d'études sous forme d'un logiciel, appelé telecrypt, de telle sorte de garantir une manipulation souple et facile des paramètres de transmission/réception de données.

Summary

This master thesis deals with the realization of a secure digital wireless communication system based on Raspberry Pi. The security of the transmission is ensured by several modes namely encryption by shift, encryption by block, as well as direct sequence spread spectrum techniques. The main objective is the construction of a wireless local network, full duplex, supporting multiuser and completely independent of the Internet. The solution proposed to realize this system is to use the famous Rasberry Pi computer. First, we configured a WiFi network in Raspberry Pi 1, by choosing the WPA2 protocol to ensure maximum protection. Then we used Raspberry Pi 4 to create a samba program that manages the files and creates the NAS server. The latter allows sharing data with other devices such as Smartphone, tablets and other computers. The most critical part of this project is how to secure the transmitted data by implementing algorithms on Python and Matlab including the three modes cited-above. In order to permit a flexible setting of the system parameters, either in transmission or reception, our work is organized as a software called *telecrypt* managed through a graphical user interface. The created GUI allows an

interactive product management permitting to use the three modes to secure text files and reconstruct them at the receiver part in an easy and simple manner.

ملخص :

تتناول أطروحة الماستر هذه إنجاز نظام اتصالات رقمي لا سلكي آمن يعتمد على اثنين من الراسب بيري باي1و 4 . يتم تحقيق أمن الارسال من خلال عدة تقنيات و هي التشفير بالإزاحة، التشفير بالكتلة و كذلك أنظمة الطيف الموسع. الهدف الرئيسي هو انشاء شبكة محلية لاسلكية آمنة مستقلة تماما عن شبكة الأنترنيت. الحل المقترح لتحقيق هذه الغاية يعتمد أولا على تكوين شبكة وايفاي براسب بيري باي1 باختيار بروتوكول امان WPA2 لتوفير أقصى حماية و بعدها العمل على راسب بيري باي4 لغرض انشاء برنامج سامبا لادارة الملفات و انشاء خادم NAS بهدف مشاركة البيانات مع العديد من الأجهزة كالحواسيب، الهواتف الذكية و غيرها. أما الخطوة التالية فتتعلق بأمن البيانات من خلال إدماج خوارزميات تتعلق بطرق التأمين الثلاث المذكورة آنفا مستعملين في ذلك برنامجي Python و Matlab. إضافة إلى كل ذلك، قررنا إنشاء برنامج يسمى telecrypt يضم واجهة تفاعلية تسهل استخدام برنامج ادارة المنتج و يضم التقنيات الثلاث لتأمين الإرسال مما يسمح بتشفير محتوى الملف النصي المرسل و يجعل مهمة بعث و استقبال المعطيات سهلة وآمنة لكل المستخدمين.

Liste des abréviations

AP: Access Point.

AES : Advanced Encryption Standard.

BSD: Berkeley Software Ditribution.

Code ASCII: American Standard Code Information Interchange.

Code PRN: Code Pseudo-Bruit Aléatoire.

Code PN: Code Pseudo- Noise.

CDMA: Code Division Multiple Access.

Cloud: Stockage en ligne.

DTE: Data Terminal Equipment

DES: Data Encryption Standard

DHCP: Dynamic Host configuration Protocol.

DNS: Domain Name System.

ETTD: Equipment Terminal de Traitement de Données.

GPS: Global Positioning System.

GPU: Graphics Processing Unit.

HDMI: High Definition Multimedia Interface.

IP: Internet Protocol.

LAN: Local Area Network.

LFSR: Linear-Feedback Shift Registers.

NAS: Network Attached Storage

OS: Operating System.

SD: Secure Digital.

USB: Universal Serial Bus.

UTF-8: Universal Character Set Transformation Format - 8 bits

UI: user interface

Ux: user experience

VCC: Common Collector Voltage.

Wi-Fi: Wireless Fidelity.

WLAN: Wireless Local Area Network.

WPA: Wi-Fi Protected Access.

XOR: exclusive OR.

Liste des figures

Chapitre1

Chapitre2

Chapitre3

Chapitre4

Table des Matières

Introduction générale

Introduction générale

Les moyens de communication se sont accrus à travers le monde grâce aux progrès technologiques et les recherches en électroniques. Un monde plein de mouvement, plein de vitalité et des richesses, l'être humain était souvent à la recherche des nouvelles qui l'entourent et qui peuvent fournir une meilleure qualité de vie et un confort qui ne connait pas de limite. Sa curiosité était souvent un paramètre primordial pour son développement. Mais comme la curiosité de l'être humain était un avantage pour sa progression et son développement, elle était aussi pour certaines personnes un paramètre indésirable par le fait du vol, l'espionnage et l'intrusion et autres comportements similaires. Donc, la technologie, par son rôle était toujours présent du fait qu'elle réduisait de ces comportements non souhaités et essayait de les éliminer pour un meilleur confort sécuritaire des personnes et leurs patrimoines.

La technologie des systèmes de sécurité a vécu une chronologie et une diversité depuis que l'homme était responsable de lui-même et de sa sécurité commençant par le recours aux moyens simples, passant par la création des mécanismes de sécurité, jusqu'à l'avènement de l'électronique. L'apparition de l'électronique dans notre vie quotidienne nous a apporté assez d'avantage, la réalisation des systèmes de sécurité électronique avec divers logiciels se caractérise par la fiabilité,

Les recherches et le développement en domaine d'électronique ont montré que chaque personne a ses propres méthodes de programmation ce qui conduit à la différence des méthodes de sécurité, donc la diversité des logiciels, et l'élévation du niveau de sécurité.

Dans notre projet de fin d'études, on a abordé le thème de réalisation d'un système de communications numériques sécurisé autour d'un d'un Raspberry pi. La sécurité de transmission est effectuée par plusieurs techniques à savoir le chiffrement par décalage, chiffrement par bloc, ainsi que l'étalement de spectre.

Dans notre projet nous avons travaillé avec deux Raspberry pi (4 et 1). Le but principal est de créer un réseau local sans fils sécurisé qui ne dépend absolument pas de la connexion Internet. Pour se faire, nous avons procédé comme suit :

1) Nous avons installé le système Raspbian dans Raspberry pi 1. Après, nous avons installé le logiciel hostapd afin de configurer par la suite le réseau WiFi (nom de point d'accès, mot de passe…). Nous avons choisi le protocole de sécurité WPA2, théoriquement impiratable et qui offre plus de protection que WPA. De plus, ceci permet d'utiliser des mots de passe allant jusqu'à 63 caractères.

2) Nous avons travaillé sur Raspberry pi 4. Celui-ci intègre le logiciel samba pour bien gérer les fichiers dans le réseau. En suite, nous avons créé le serveur NAS (Network Attached Storage), en se servant d'un flash disque USB pour le sauvegarde des données. Cependant, il est à noter que le Raspberry pi 4 contient quatre ports USB et donc peut supporter quatre disques durs externes soit une capacité de stockage de NAS de 4 Tb au moins.

L'utilisation d'un serveur NAS comporte de nombreux avantages :

- Il permet de centraliser ses données et de les partager avec plusieurs appareils (convergence numérique).
- Il permet de sauvegarder l'ensemble de ses données à partir d'un endroit unique.
- Il permet de rester propriétaire de ses données.

3) L'étape suivante concerne la sécurité des données. Nous avons travaillé avec les langages Python et Matlab. Trois modes de cryptage ont été utilisés:

- Mode César
- Mode AES
- Le cryptage par étalement de spectre

4) Nous avons créé une interface graphique pour simplifier l'utilisation du logiciel gestionnaire du réseau sans fils réalisé. Nous appliquons UX/UI du logiciel Qt designer pour la création d'une interface graphique interactive facilitant l'utilisation du logiciel de gestion réalisé. Ce dernier nous permet d'utiliser les trois modes cités plus avant pour crypter un fichier texte. Ceci rend le décodage une tache pratiquement impossible.

Pour décrypter le fichier, il faudra:

- Connaitre les modes appliquées
- Connaitre l'ordre des modes appliquées
- Connaitre les paramètres des clés utilisés à savoir : Décalage de César ; clé, iv et mode utilisé de l'AES et les paramètres n et L de la séquence pseudo-aléatoire utilisé dans l'étalement de spectre à séquence directe. Par conséquent, six paramètres sont requis pour pouvoir déchiffrer un fichier émis.

Mise à part l'introduction générale et la conclusion, ce document présente une étude théorique et pratique organisée en quatre chapitres selon la chronologie suivante :

1- La partie théorique se compose de :

- Chapitre I : Généralités sur les transmissions numériques sécurisées.

2- La partie pratique est composée de :

- Chapitre II : Analyse structurelle du système de transmission sécurisée réalisé.
- Chapitre III : Analyse algorithmique du système de transmission sécurisée réalisé.
- Chapitre IV : Résultats et discussions.

Chapitre 1 :Généralités sur les transmissions numériques sécurisées

I .1 Introduction:

De nos jours, communiquer de n'importe quel lieu et avec n'importe qui n'est plus considéré comme un luxe mais fait partie courante de notre vie. Depuis l'avènement de l'ère des transmissions numériques, ces communications sont devenues accessibles à tous.

La transmission de données entre un émetteur et un récepteur suppose que soit établie une liaison sur un support de transmission (appelée aussi voie de transmission ou canal) munie d'équipement de transmission à ses extrémités [1].

La transmission de données doit être sécurisée pour rendre le réseau privé à l'abri de tout accès non autorisé. Plusieurs méthodes de cryptographie comme le code César et AES permettent de réaliser la sécurité souhaitée. De plus, un étalement de spectre à séquence direct est aussi introduit ce qui renforce la sécurité de transmission dans le réseau sans fils réalisé.

I.2 Système de base d'une transmission numérique:

Une transmission numérique permet de transmettre une information numérique, c'est-à dire une suite de''0'' et ''1'', d'un émetteur à un récepteur via un canal de transmission. Le schéma de principe d'une chaine de communication numérique est présenté sur la Figure 1.1, un signal numérique n'existe pas de point de vue physique, il est nécessaire de le transformer en un signal analogique afin de le transmettre via le canal. Ce canal de transmission permet de transmettre un signal d'un point A à un point B, mais comme tout canal de transmission il va engendrer des erreurs sur le signal. Le signal qui sera reçu au point B ne correspondra pas exactement au signal qui a été émis en A. Afin de lutter contre ces erreurs de transmission, l'émetteur est équipé d'un bloc de codage canal et le récepteur d'un bloc de décodage. Au niveau de l'émetteur, ce bloc a pour rôle de rajouter de la redondance au message qui doit être émis. Cette redondance va permettre, au niveau du récepteur, de détecter et/ou de corriger d'éventuelles erreurs qui seraient intervenues lors de la transmission du message. Dés lors, toute transmission numérique moderne, quelle soit filaire, terrestre ou satellitaire, est équipé d'un bloc de codage de canal afin de lutter contre les erreurs de transmission [2].

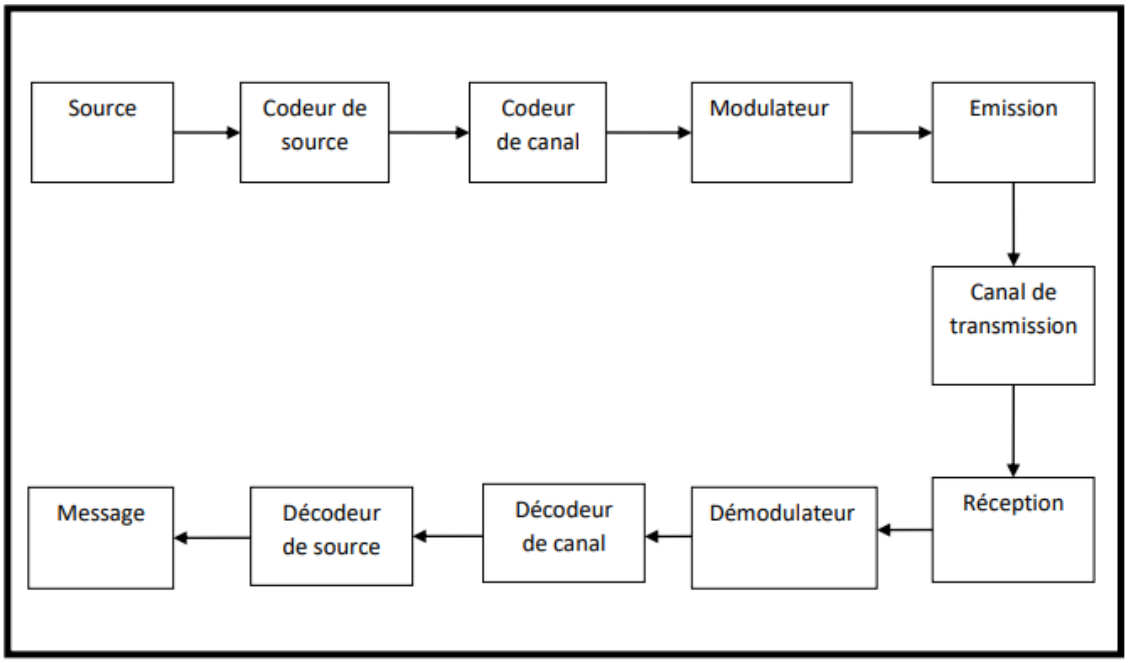


Figure 1.1 : Synoptique d'une transmission numérique.

L'information binaire n'arrive pas toujours intacte au destinataire, et les performances du système de transmission dépendent de très nombreux facteurs, parmi lesquels on peut citer les caractéristiques du canal, la puissance de l'émetteur, la forme d'onde utilisée ou encore le type de codage.

Pour améliorer la qualité et la portée d'une transmission numérique, on peut accroître le rapport signal à bruit, c'est-à-dire, par exemple, augmenter la puissance émise et/ou diminuer le facteur du bruit du récepteur [2].

I.3 Caractéristiques des supports de transmission [3]

Les supports de transmission exploitent les propriétés de conductibilités des métaux (paires torsades, câble coaxial) ou celles des ondes électromagnétiques (faisceau hertzien, fibre optique). Les caractéristiques essentiels des supports de transmission sont :

- **Affaiblissement :** Un canal de transmission atténue (affaiblit) l'amplitude du signal qui le traverse.
- **Le déphasage :** Le déphasage encore appelé distorsion de phase, implique un retard du signal reçu par rapport au signal émis dû au temps de propagation de ce signal de l'émetteur vers le récepteur.
- **La bande passante :** Le support de transmission se comporte généralement comme un filtre qui ne laisse donc passer qu'une bande limitée de fréquence appelée bande passante.

- **Le bruit :** Le bruit est un signal perturbateur provenant du canal lui-même ou de son environnement externe. Il est de comportement aléatoire est vient s'ajouter au signal véhiculant les informations et provoque ainsi les erreurs de transmission.

I.4 Techniques de transmission du signal numérique

Les réseaux informatiques se fondent sur la numérisation des informations, c'est à dire la représentation des données par des suites de ' 0 ' et de ' 1 '. Ils englobent la transmission de ces données, leur mémorisation dans des mémoires de stockage et enfin leur utilisation. La première étape consiste donc à ramener les informations que nous voulons échanger à un ensemble d'informations binaires à l'aide de techniques de codage. Pour cela, on utilise des codes, qui font correspondre à chaque caractère une suite précise d'éléments binaires ou bit.

Pour transmettre ces informations binaires sur un canal de transmission, il est nécessaire de les transformer au préalable en un signal électrique. La méthode la plus simple consiste à représenter l'élément binaire ' 0 ' par une tension V0, et l'élément binaire '1' par une tension V1. Une modulation du signal de source s'avère nécessaire pour pouvoir transmettre l'information à longues distances.

I.5 Transmissions numériques sécurisées

I.5.1 Définition de cryptographie:

La cryptographie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse [4]. Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles Sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préférera le verbe (chiffrer).

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 [5].

I.5.2 Différence entre chiffrement (ou cryptage) et codage [5]

Les opérations de chiffrement et du codage font partie de la théorie de l'information. La différence essentielle réside dans la volonté de protéger les informations et d'empêcher des tierces personnes d'accéder aux données dans le cas du chiffrement. Le codage consiste à transformer de l'information (des données) vers un ensemble de mots. Chacun de ces mots est constitué de symboles. La compression est un codage : on transforme les données vers un ensemble de mots adéquats destinés à réduire la taille mais il n'y a pas de volonté de dissimuler (bien que cela se fasse implicitement en rendant plus difficile d'accès le contenu).

Le « code » dans le sens cryptographique du terme travaille au niveau de la sémantique (les mots ou les phrases). Par exemple, un code pourra remplacer le mot « avion » par un numéro. Le chiffrement travaille sur des composants plus élémentaires du message, les lettres ou les bits, sans s'intéresser à la signification du contenu. Un code nécessite une table de conversion, aussi appelée « dictionnaire » (code book en anglais). Ceci étant, « code » et « chiffrement » sont souvent employés de manière synonyme malgré cette différence.

On peut aussi considérer que le chiffrement doit résister à un adversaire « intelligent » qui peut attaquer de plusieurs manières alors que le codage est destiné à une transmission sur un canal qui peut être potentiellement bruité. Ce bruit est un phénomène aléatoire qui n'a pas « d'intelligence » intrinsèque mais peut toutefois être décrit mathématiquement.

I.5.3 Méthodes de sécurisation de l'information utilisées dans ce projet

a. Chiffrement par décalage « code de César » :

- **Définition :**

Le chiffrement par décalage de César compte parmi les plus élémentaires des chiffrements par substitution -- ceux qui consistent à remplacer les lettres du texte clair par les lettres correspondantes de l'alphabet du texte chiffré.

Bien que cette méthode n'ait certainement pas été inventée par le célèbre empereur, elle lui doit son nom en raison de ce passage de la Vie des douze Césars de Suétone : « On possède enfin de César des lettres à Cicéron, et sa correspondance avec ses amis sur ses affaires domestiques. Il y employait, pour les choses tout à fait secrètes, une espèce de chiffre qui en rendait le sens inintelligible (les lettres étant disposées de manière à ne

jamais pouvoir former un mot), et qui consistait, je le dis pour ceux qui voudront les déchiffrer, à changer le rang des lettres dans l'alphabet, en écrivant la quatrième pour la première, c'est-à-dire le d pour le a, et ainsi de suite ».

Un chiffrement par décalage de César se perce plus ou moins facilement selon l'information dont vous disposez. Si vous connaissez la technique employée, il vous suffira d'attaquer le message par force brute en testant les transformations ROTN jusqu'à trouver la bonne (voir ci-dessous) [6].

- **Principes du chiffrement de César:**

Le chiffrement s'effectue très facilement, en listant les lettres de l'alphabet les unes à la suite des autres, puis en plaçant au-dessous ces mêmes lettres décalées d'un certain nombre de rang donné. Le décalage évoqué par Suétone dans sa biographie donnerait donc lieu à ce tableau :

Pour chiffrer un message, il suffit alors de remplacer la lettre prise dans la première ligne par la lettre correspondante dans la seconde. Ainsi, « Prends garde aux Ides de Mars » se transforme en « Suhqgv jdugh dxa Lghv gh Pduv ».

Pour déchiffrer le message, son destinataire ne doit connaître que le nombre de positions dont l'alphabet a été décalé. Ce type de transformation se nomme ROTN, où N désigne le nombre de positions en question. Avec un alphabet décalé de trois places, le décalage de César est donc une transformation ROT3.

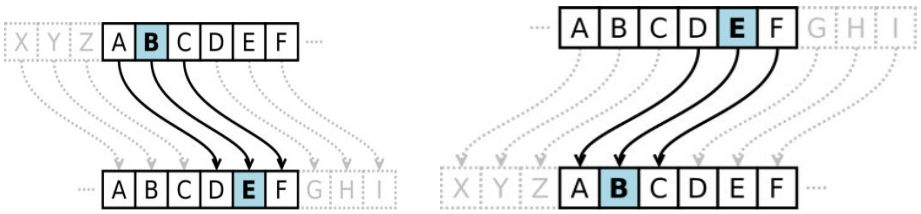


Figure 1.2: Chiffrement et déchiffrement par décalage [7].

b. Chiffrement par bloc « AES » :

- **Définition :**

L'AES (Advanced Encryption Standard) est, comme son nom l'indique, un standard de cryptage symétrique destiné à remplacer le DES (Data Encryption Standard) qui est devenu trop faible au regard des attaques actuelles.

La norme AES a été adoptée dans une gamme plus grande initiée par Rijndael – un nom composé des deux noms de ses créateurs belges. Il a été analysé en profondeur et est maintenant utilisé dans le monde entier par certaines des applications de sécurité les

plus exigeantes. AES est connu comme un bloc de cryptage symétrique, ce qui signifie qu'il crypte et décrypte les données en blocs de 128 bits chacun. Pour ce faire, il utilise une clé cryptographique spécifique, qui est effectivement un ensemble de protocoles pour masquer les informations. Cette clé peut avoir une taille de 128, 192 ou 256 bits [8].

c. Étalement de spectre [9]

- Les transmissions à spectre étalé, initialement élaborées pour une utilisation militaire, sont entrées depuis quelques années dans un grand nombre d'application civiles et tout particulièrement depuis l'émergence de la technique d'accès multiples de type CDMA (Code Division Multiple Access). Cette technique utilise l'étalement de spectre par séquence directe afin de permettre la transmission simultanée de signaux issus de plusieurs utilisateurs à l'intérieur d'une même bande de fréquence, tout en assurant un taux d'interférences inter-utilisateurs assez faible grâce à un type particulier de codes ayant des propriétés de corrélation spécifiques. Ces séquences ont une longueur très grande par rapport à celle du message basse fréquence à transmettre d'où le terme étalement de spectre.
- L'étalement de spectre est une technique où un signal est transmis sur une bande passante considérablement plus large que l'ensemble des fréquences composant le signal original ne serait transmis par des méthodes classiques de modulation. Cette technique diminue le risque d'interférences avec d'autres signaux reçus tout en garantissant une certaine confidentialité [10].
- L'étalement de spectre utilise généralement une séquence pseudo-aléatoire (ou PN pour pseudo noise) créé par des circuits logiques très simples à base de portes xor pour étaler le spectre du signal d'information à bande étroite. Ceci permet de brouiller l'information émise. Cette opération dite Scrambling en anglais est d'aussi efficace que la longueur $L=2^n-1$ du code PN est longue ; n étant le nombre de registres à décalage. Le récepteur récupère le signal en corrélant le signal reçu avec une réplique de cette séquence.
- Les transmissions à spectre étalé, initialement élaborées pour une utilisation militaire, sont entrées depuis quelques années dans un grand nombre d'applications civiles et tout particulièrement depuis l'émergence de la technique d'accès multiples de type CDMA. Le domaine militaire est l'origine de la technique par l'étalement de spectre car il nécessite des communications indétectables et non-brouillées ainsi

qu'après on l'a utilisé dans le domaine civil pour la mise en œuvre de systèmes de communications robustes [11]. Parmi les principaux domaines, nous citons :

- ✓ La radiolocalisation GPS…
- ✓ Les liaisons par satellites,
- ✓ Les réseaux cellulaires radio mobiles,
- ✓ Les réseaux locaux sans fil,
- ✓ Les applications industrielles, scientifiques et médicales,
- ✓ Les transmissions par courant porteur,
- ✓ Le tatouage de l'information et des images.

- **PN code** : Une pseudo-noise code (code PN) ou code pseudo-bruit aléatoire (code PRN) est un code qui a un spectre similaire à une séquence aléatoire de bits mais qui est généré de manière déterministe. Les séquences les plus couramment utilisées dans les systèmes à spectre étalé à séquence directe sont les séquences de longueur maximale, les codes Gold, les codes Kasami et les codes Barker.

Les séquences maximales : Une séquence maximale (Figure 1.3) est une séquence périodique pour laquelle la longueur L de la période est maximale pour le nombre *n* de bascules du registre à décalage et vaut $L = 2^n - 1$. La longueur *L* représente le nombre des chips dans une période. On peut générer une séquence maximale uniquement lorsque le nombre de prises, excluant la prise de rétroaction [12].

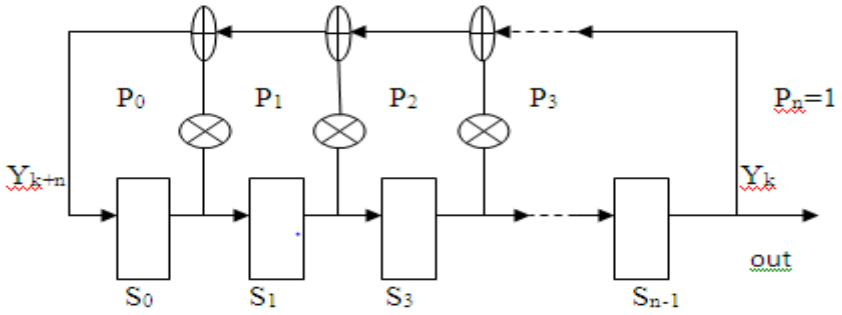


Figure 1.3: Paramètres d'entrée et de sortie d'un générateur LFSR [13].

Codes Gold : Un code de Gold est un type de séquence binaire, et sont nommées d'après Robert Gold. Les codes de Gold sont bornés par des petites corrélations croisées au sein d'un ensemble, ce qui est utile lorsque plusieurs périphériques diffusent dans la même gamme de fréquences. Un ensemble de séquences de codes Gold se compose de 2^n-1, chacune des séquences avec une période de 2^n-1 [14]. Les codes Gold sont générés à partir de deux séquences maximales formant impérativement un pair préféré.

I.6 Conclusion

Nous avons, dans le présent chapitre, donné un aperçu rapide sur les transmissions numériques et décrit les méthodes de sécurisation de données utilisés dans notre projet. Ces méthodes se divisent en deux grandes catégories : 1) cryptographie et 2) étalement de spectre. Les trois modes de sécurisation sont appliqués sur le message de source qui véhiculera sur l'air libre avant d'atteindre le récepteur distant à travers une liaison WiFi. Le chapitre qui suit présente l'analyse structurelle du système de transmission numérique sécurisé expliquant la conception matérielle proposée relative à notre projet.

Chapitre 2 : Analyse structurelle du système de transmission sécurisé réalisé

II.1 Introduction:

Pour la réalisation de projets dans quel domaine, une étude théorique est nécessaire et qui sera suivie par une concrétisation pratique.

L'implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influe énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser.

Dans ce chapitre, nous allons commencer par la description de l'environnement de travail, schéma synoptique puis à dégager et élaborer les composantes de notre système y compris le Raspberry pi ; le cœur du système de communication conçu.

II.2 Schéma synoptique:

Le schéma synoptique du projet réalisé et rapporté sur la Figure 2.1.

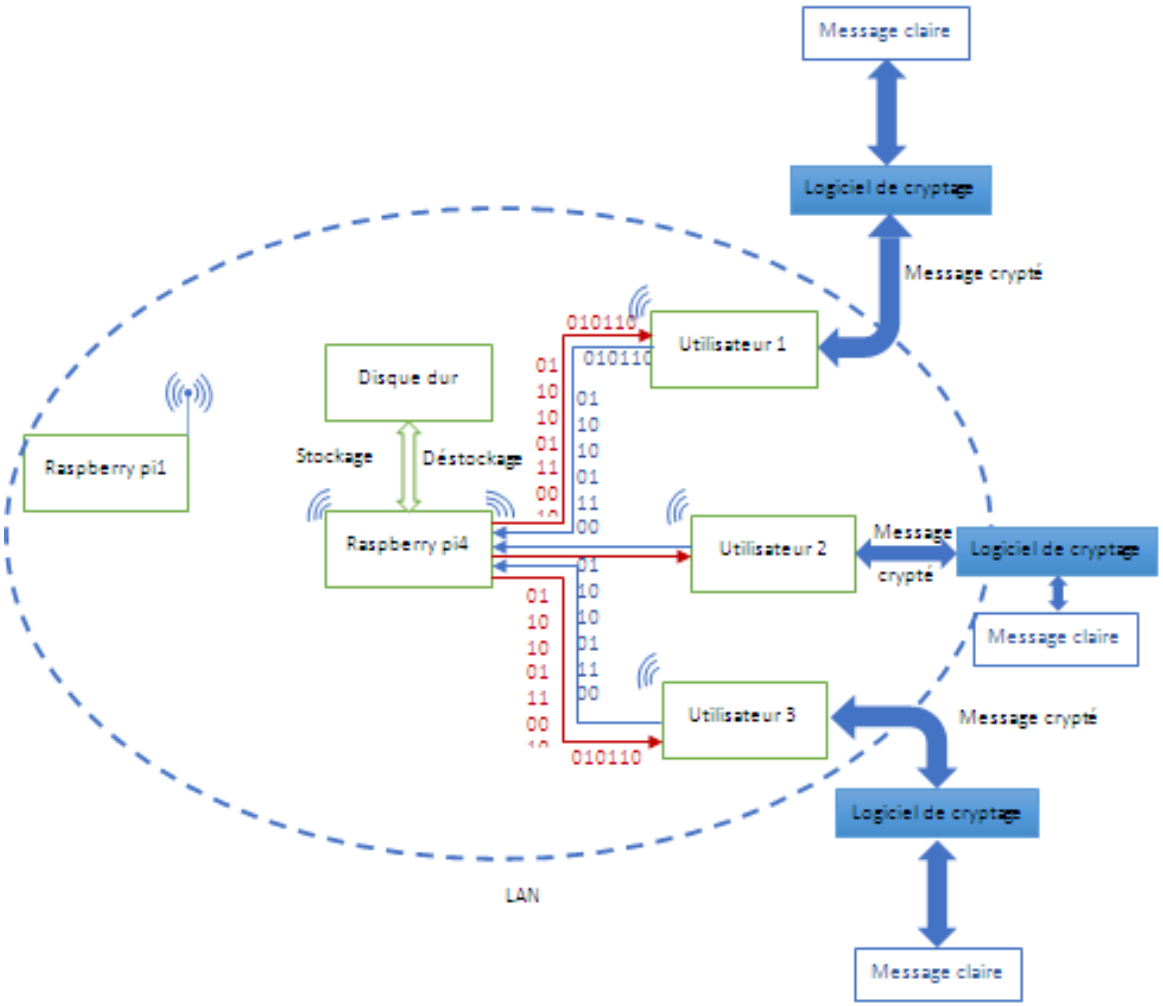


Figure 2.1: Schéma synoptique du projet réalisé.

II.3 Principe de fonctionnement:

Notre projet est divisé en trois parties (voir Figure 2.2) :

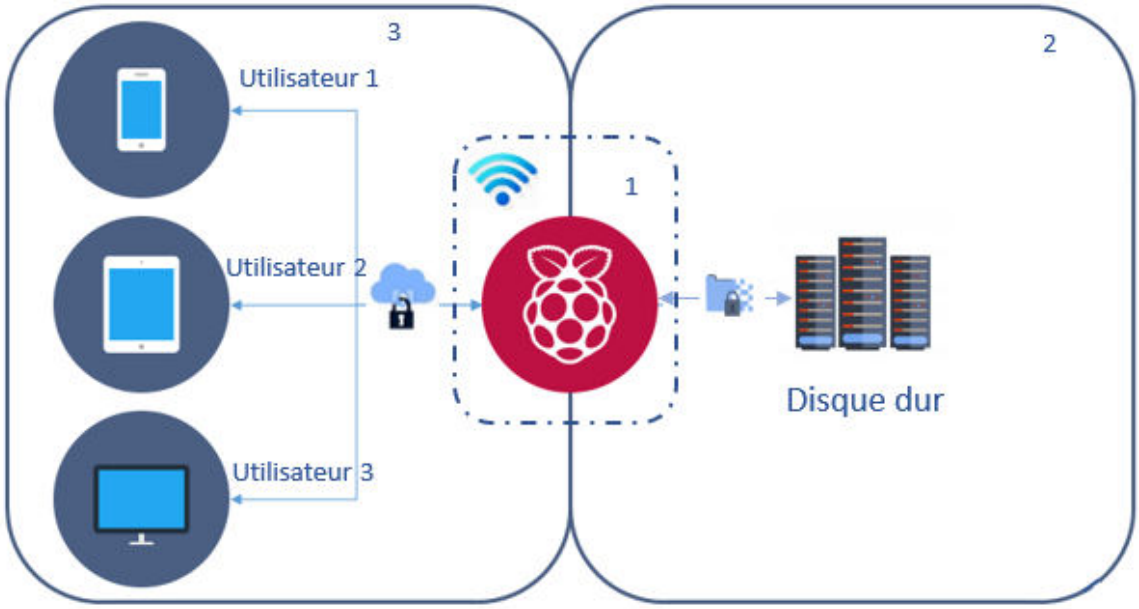


Figure 2.2 : Principe de fonctionnement du projet réalisé.

- **Partie 1 :**

Point d'accès : On a besoin de mettre en place un **point d'accès** Wi-Fi avec notre Raspberry Pi. La solution qu'on a adoptée est **hostapd**.

Accessoirement, il faut installer un serveur **DHCP** pour distribuer des adresses aux machines qui se connectent.

- **Partie 2 :**

Gestion et transfert des fichiers : Le logiciel **Samba** est utilisé pour générer et partager les fichiers à travers le réseau en milieu hétérogène supportant des systèmes d'exploitation divers à savoir Windows, Linux, Android ….

- **Partie 3 :**

Chiffrements des fichiers : Pour sécuriser notre fichier, on a utilisé trois modes à savoir le mode AES, le mode César et la technique d'étalement de spectre

II.4 Définition du raspberry Pi:

Raspberry (Figure 2.3) est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch...). Elle est aussi

capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles [15].

Figure 2.3 : Raspberry Pi4.

II.4.1 Caractéristiques du Raspberry Pi utilisé:

Le Raspberry Pi est un nano-ordinateur de la taille, grosso modo, d'une carte de crédit. Ce qui est impressionnant, ce sont ses caractéristiques. Sur un petit bout de carte électronique, il y a un [16]:

- Processeur 1.5GHz quad-core 64-bit ARM Cortex-A72.
- 1GB, 2GB, ou 4GB de mémoire SDRAM LPDDR4.
- Ethernet Gigabit.
- Wifi Dual-band 802.11ac.
- Bluetooth 5.0.
- Deux ports USB 3.0 et deux ports USB 2.0.
- Support double écran.
- GPU VideoCore VI.

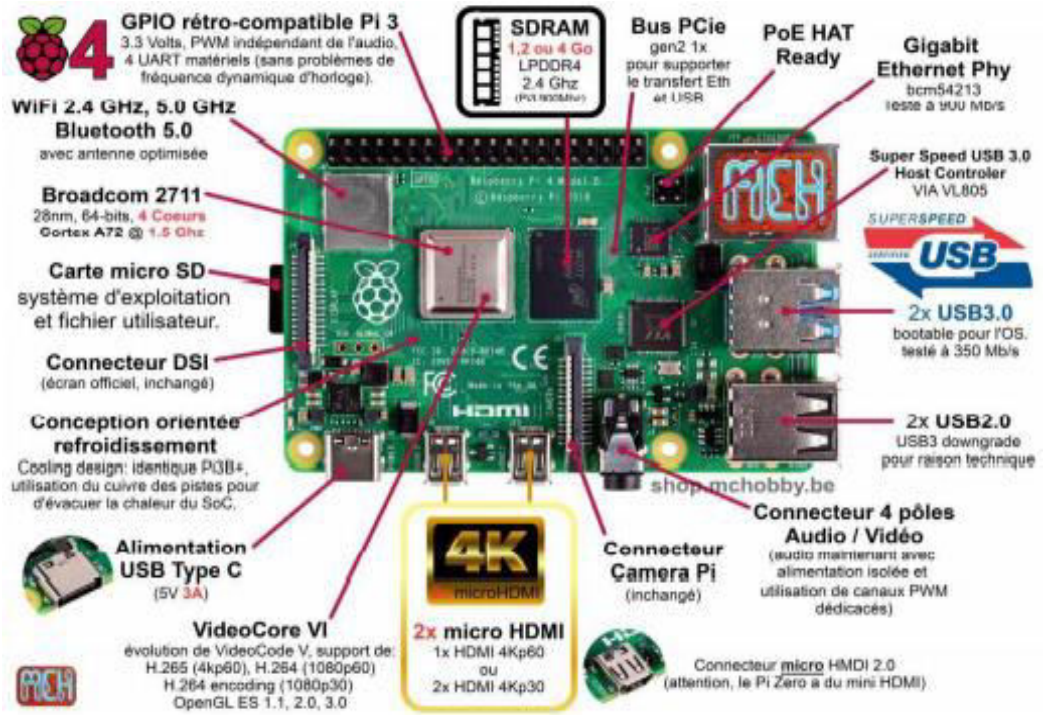


Figure 2.4 : Fiche technique de Raspberry Pi 4 [16].

II.4.2 Accessoires du Raspberry Pi [17][18]:

- **Alimentation :**

  Le Raspberry Pi s'alimente sous tension unique de 5 volts, tension sur laquelle il peut consommer jusqu'à 1.8A selon les taches qu'il exécute.

- **Carte microSD :**

  Pour héberger notre système d'exploitation (Raspbian) nécessaires au bon fonctionnement notre Raspberry Pi, et tous les fichiers nous auront besoin d'espace de stockage c'est pour cela on a choisit une capacité de 16 Go pour notre projet.

- **Un clavier et une souris :**

  Pour commencer à utiliser votre Raspberry Pi et la première configuration.

- **Un écran de télévision ou d'ordinateur :**

  Nous avons besoin d'un écran et d'un câble pour relier l'écran et le Raspberry Pi.

- **broches de connexion :**

  Vcc : alimentation +5V DC.

- **modem :**

  Le modem (mot-valise, pour modulateur-démodulateur), est un périphérique servant à communiquer avec des utilisateurs distants par l'intermédiaire d'un réseau analogique

(comme une ligne téléphonique).On utilise le modem pour le téléchargement des commandes.

- **webcam :**

  Nous avons utilisé une caméra pour réaliser une vidéo en temps réel.

- **Une carte réseaux USB :**

  Une carte réseau est un périphérique informatique qui fait le lien entre l'ordinateur dans lequel elle est installée et le réseau auquel elle le connecte. Elle est constituée d'un ensemble de composants électroniques soudés entre eux sur un même circuit imprimé. Plusieurs d'objets sont connectés aux réseaux: ordinateurs, Smartphones, tablettes, consoles, imprimantes, télévisions.

II.4.3 Configurations [19][20]:

Comme tous les autres ordinateurs, le Raspberry Pi utilise un système d'exploitation. Celui fourni par défaut est une version de GNU/Linux appelée Raspbian qui est idéal pour l'esprit de partage qui règne au sein de la la communauté Raspberry Pi car il est libre 4 et open source 5. Ce logiciel à été écrit dans le cadre d'un projet communautaire pour ceux qui recherchent une alternative au monopole de Micrrosoft Windows et d'Apple OS X. Linux est un système d'exploitation Open Source. c'est un système d'exploitation complet, basé sur les mêmes concepts robuste d'UNIX qui est apparu au début de l'informatique. Il a de nombreux partisans fidèles et serviable et s'est transformer en un système d'exploitation puissant et facile à utiliser.

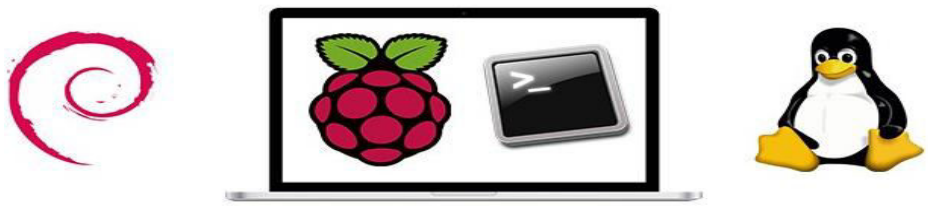


Figure 2.5 : Rasbian + lunix.

- **NOOBS :** Lorsque l'on débute avec son Raspberry Pi, on se retrouve rapidement perdu lorsqu'il faut choisir un système d'exploitation sachant que sous Linux, il existe de nombreuses distributions différentes. Heureusement, le logiciel « NOOBS » est prévu pour tester facilement les systèmes d'exploitation classiques que l'on retrouve couramment sur Raspberry Pi. C'est aussi un assistant d'installation développé par Raspberry Pi Foundation, destiné à simplifier l'installation des utilisateurs débutants dans le secteur Linux.

- **Raspbian** : C'est la célèbre distribution classique basée du **Debian**. Ce terme est un mot-valise formé par la fusion des mots « **Raspberry Pi** » et « **Debian** ». Il s'agit d'un système d'exploitation GNU/Linux spécialement conçu et optimisé pour le Raspberry Pi (Figure 2.6).

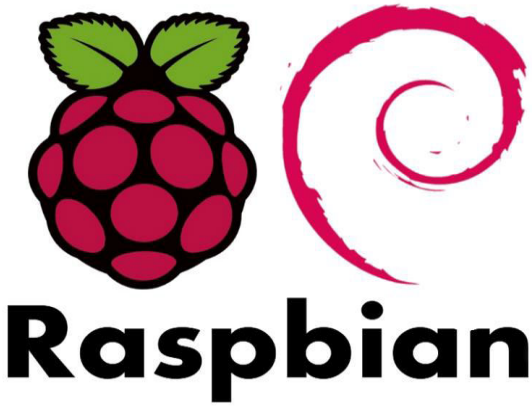


Figure 2.6: Raspberry Pi + Debian = Raspbian.

La configuration de Raspberry Pi prend en compte les tâches citées ci-après. Dans un premier temps, l'utilisateur doit fournir une carte microSD avec Raspbian téléchargé. De plus, un clavier et une souris sont nécessaires pour introduire les différentes commandes. Un écran est aussi indispensable pour visualisation en plus d'un modem afin de télécharger les logiciels Samba et hostapd. Ceci fourni, l'utilisateur doit suivre les étapes suivantes pour démarrer son Raspberry Pi pour la première fois :

1) Insérer la carte SD dans son support sur Raspberry Pi.
2) Brancher un clavier USB et une souris au Raspberry Pi.
3) Connecter la sortie micro HDMI à votre téléviseur ou moniteur.
4) Brancher l'alimentation du Pi.
5) Brancher le modem à travers le câble RJ45.
6) En cas de l'utilisation de Raspberry Pi 1, il est nécessaire d'ajouter une carte réseau USB. Celle-ci doit être configurée avant utilisation. Notons que le Raspberry Pi 4 possède sa propre carte réseau embarquée.

II.5 Logiciels associés à la configuration matérielle

Dans notre projet, les logiciels associés à Raspberry Pi sont:

- **HostAP Daemon :**

  Hostapd s'appuie sur les protocoles IEEE 802.11 AP et IEEE 802.1X/WPA/WPA2/EAP/RADIUS authentificateur.

  Hostapd permet la création d'un point d'accès WiFi, technologie sans fil utilisée pour se connecter à un réseau informatique. Dans les réseaux informatiques, un point d'accès sans fil (spot ou AP) est un dispositif qui relie les appareils de communication sans fil pour former un réseau sans fil. Le spot WiFi se connecte généralement à un réseau câblé, et peut Transmettre des données entre les appareils sans fil et les périphériques câblés. Plusieurs spots peuvent être liés ensemble pour former un réseau plus large qui permet le "roaming" (l'itinérance). Pour rappel, en revanche, un réseau où les machines clientes gèrent elles-mêmes - sans avoir besoin de point d'accès - devient un réseau ad-hoc [21].

- **Samba :**

  Le logiciel **Samba** est un outil permettant de partager des dossiers et des imprimantes à travers un réseau local. Il permet de partager et d'accéder aux ressources d'autres ordinateurs fonctionnant avec des systèmes d'exploitation Microsoft® Windows® et Apple® Mac OS® X, ainsi que des systèmes GNU/Linux, *BSD et Solaris dans lesquels une implémentation de Samba est installée.

  Pour partager de manière simple des ressources entre plusieurs ordinateurs, l'utilisation de Samba est conseillée [22].

II.6 Conclusion

Le cœur de notre système de transmission numérique sécurisé est bien évidemment les deux Raspberry Pi utilisés. Pour cela, nous avons présenté au cours de ce chapitre les composantes matérielles requises pour la configuration matérielle des Raspberry Pi ainsi que les logiciels associés. Le chapitre 3 est réservé à la description de l'analyse algorithmique du projet réalisé permettant l'émission et la réception de données textuelles à travers le réseau sécurisé.

Chapitre 3 : Analyse algorithmique du système de transmission sécurisée réalisé

III.1 Introduction

Dans ce chapitre, nous allons présenter en détails les organigrammes relatifs à notre système sécurisé de transmission numérique. En suite, la configuration matérielle des Rasbperry Pi 1 et 4 est décrite étape par étape. Les codes source sont aussi fournis pour qu'ils puissent être reproduits par le lecteur intéressé. Les programmes sont essentiellement rédigés en langage Python car il est intégré dans le système d'exploitation Linux.

III.2 Organigrammes

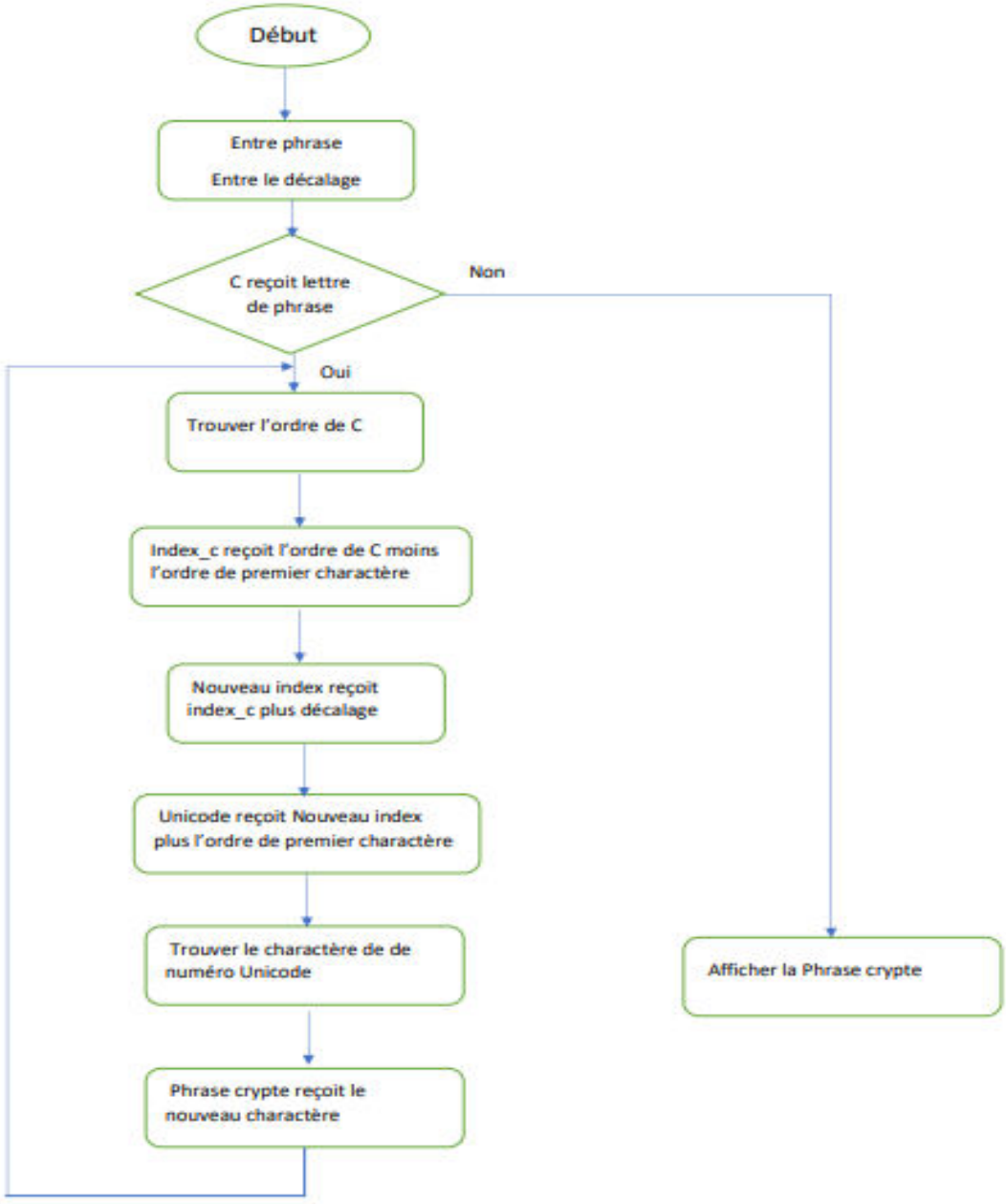


Figure 3.1: Organigramme de chiffrement César.

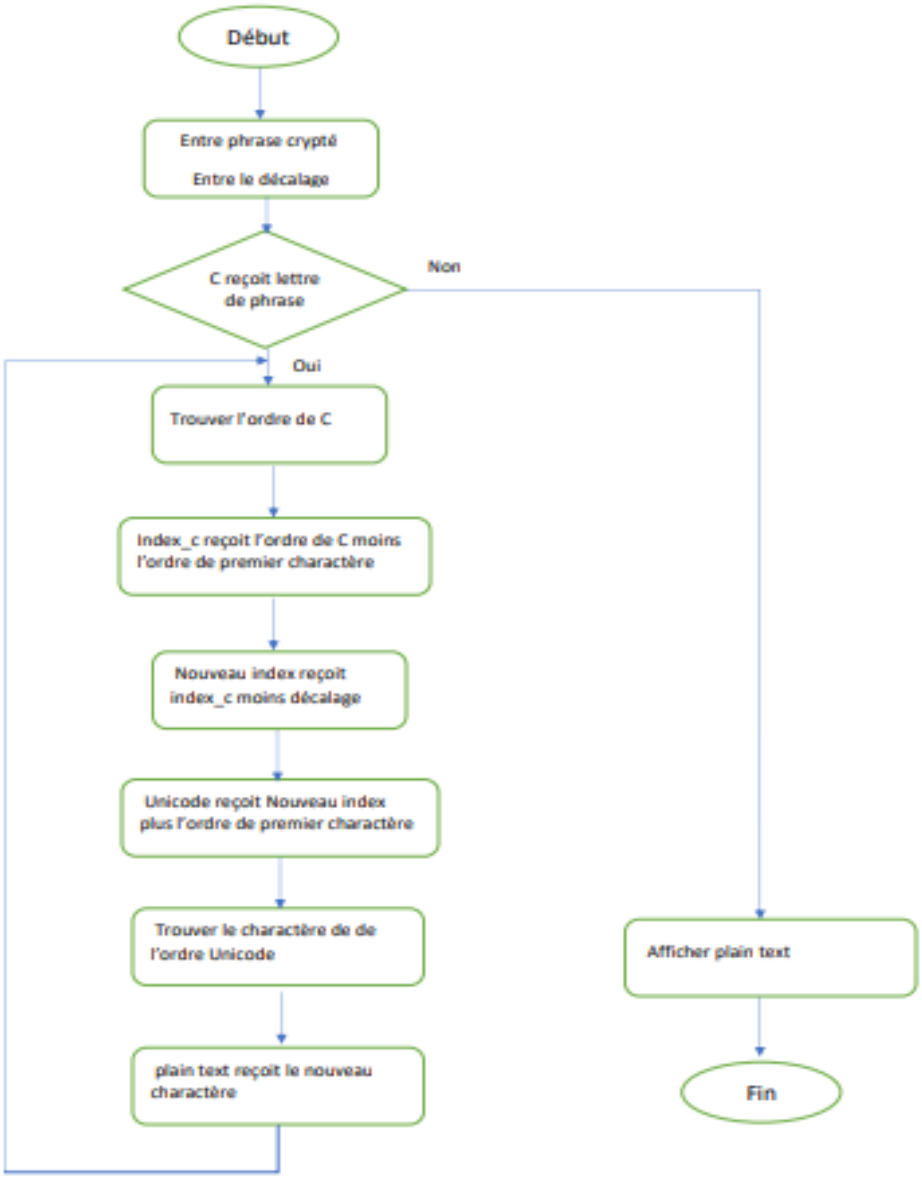


Figure 3.2: Organigrammes de déchiffrement César.

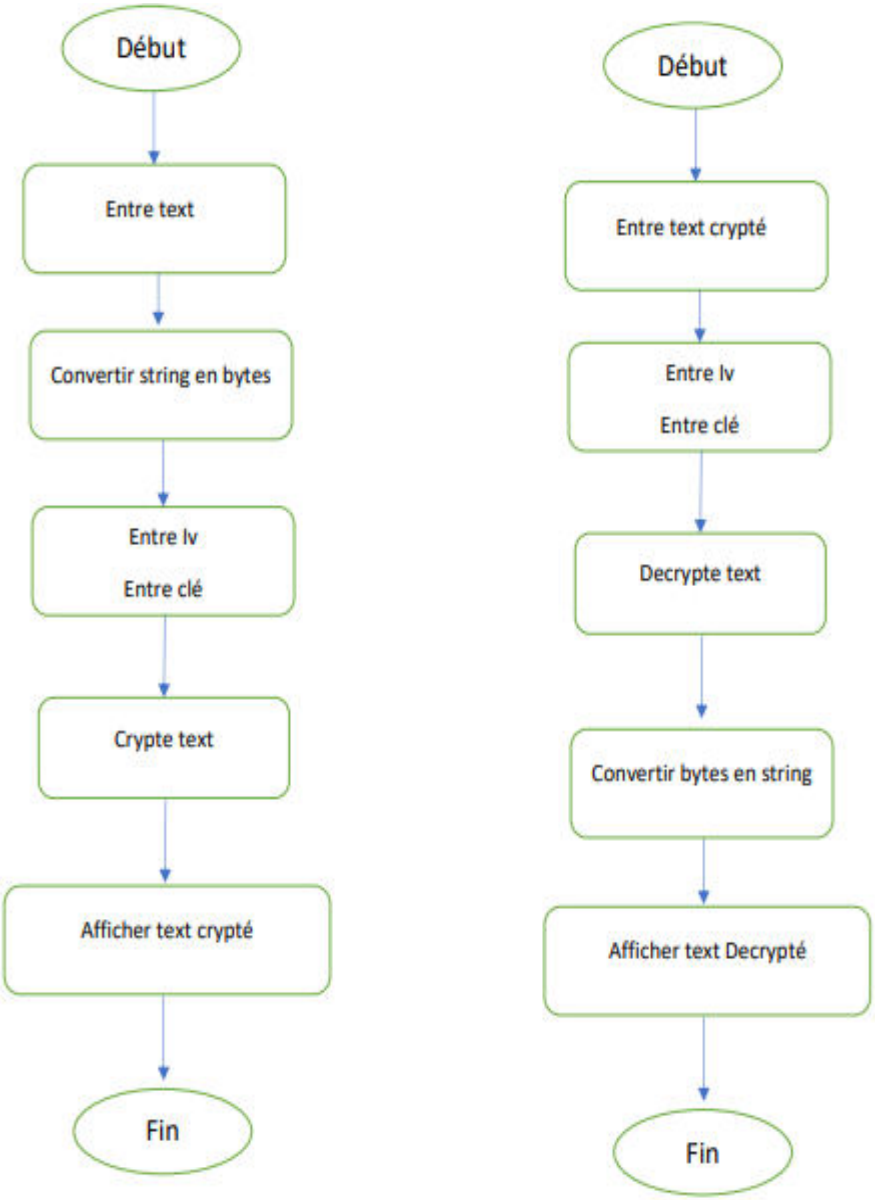


Figure 3.3: Organigrammes de chiffrement et déchiffrement AES.

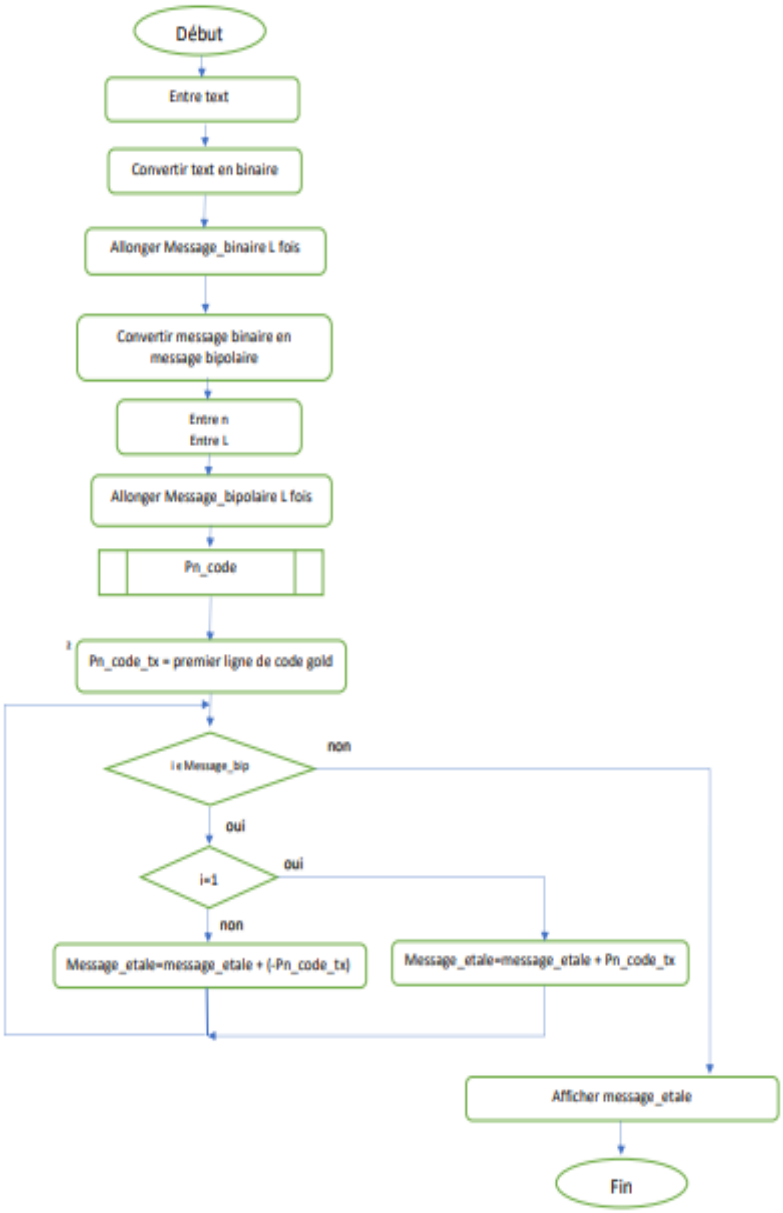


Figure 3.4: Organigramme de l'étalement de spectre par le code Gold.

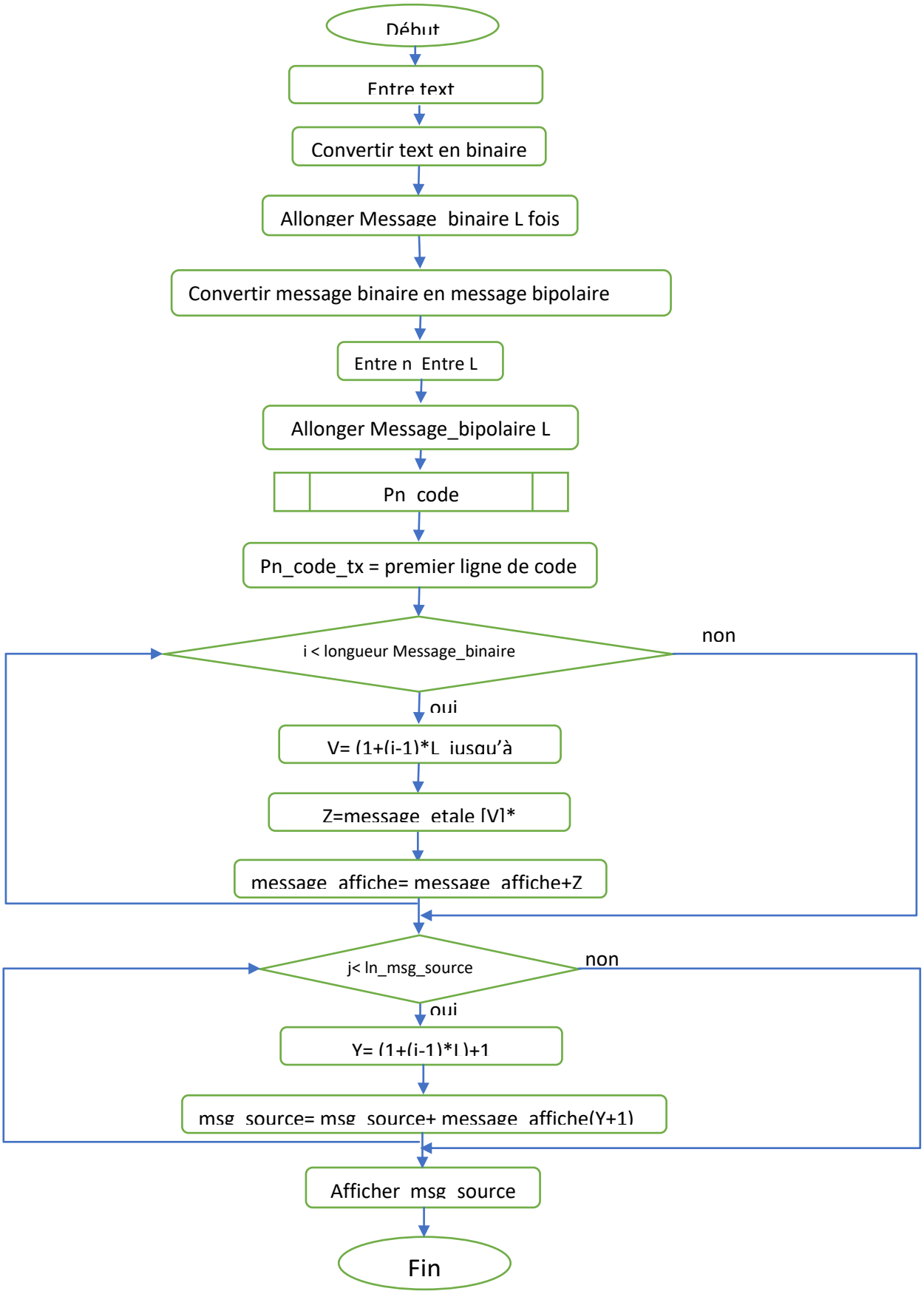


Figure 3.5: Organigrammes de l'étalement de spectre inverse utilisant le code Gold.

III.4 Logiciels associés à l'implémentation algorithmique

III.4.1 Langage Python[23]:

Python (voir le logo de la Figure 3.6) est un langage de script de haut niveau, structuré et open source. Il est multi- paradigme et multi-usage. Développé à l'origine par Guido van Rossum en 1989, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde. Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative ; c'est d'ailleurs une des raisons qui lui vaut son appellation de « langage agile ».

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en C, C^{++} ou Fortran. Il est également apprécié pour la clarté de sa syntaxe, ce qui l'oppose au langage Perl.

Figure 3.6: Logo du Python.

III.4.2 Qt designer :

Qt designer, dont le logo est présenté dans la Figure 3.7, est une série d'outils intégrés dans Qt Creator permettant de créer graphiquement des interfaces graphiques de l'application. Lors de la compilation elles seront automatiquement reconverties en python et donc utilisables comme des classes normales [24].

Figure 3.7 : Le logo de Qt designer.

III.4.3 Logiciel Matlab :

Matlab est un logiciel commercial de calcul interactif. Son logo est présenté dans la Figure 3.8. Il permet de réaliser des simulations numériques basées sur des algorithmes d'analyse numérique et un langage de développement informatique particulièrement dédié aux applications scientifiques, Il peut donc être utilise pour la résolution approchée d'´équations différentielles, d'´équations aux d´dérivées partielles ou de systèmes linéaires, il est utilisé pour développer des solutions nécessitant une très grande puissance de calcul [25].

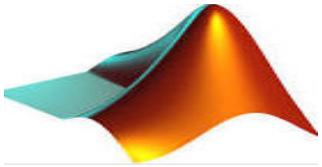


Figure 3.8 : Le logo de Matlab.

III.5 Codes sources relatives au projet réalisé:

III.5.1 Configurations relatives au Raspberry Pi 1:

a. Installation et configuration de carte réseau WiFi (Figure 3.9)

sudo apt-get update

sudo apt-get upgrade

sudo nano /etc/dnsmasq.conf

sudo nano /etc/network/interface

sudo ifconfig wlan0 192.168.42.10

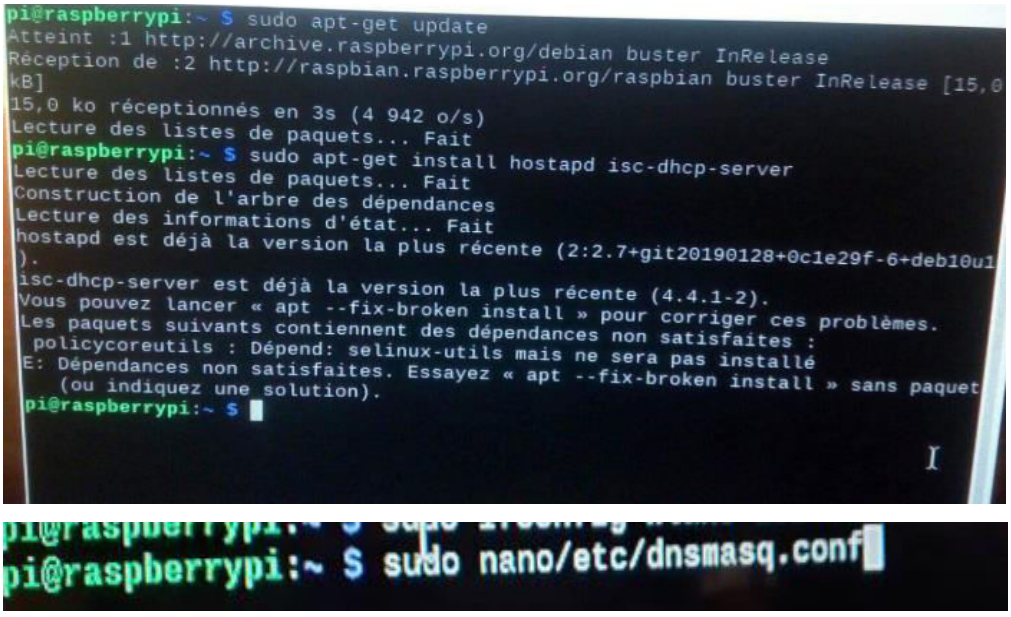


Figure 3.9: Installation et configuration de carte réseau WiFi.

b. Création de réseau hotspot (Figure 3.10)

sudo apt-get install hostapd isc-dhcp-server

sudo nano /etc/dhcp/dhcpd.conf

sudo nano /etc/default/isc-dhcp-server

sudo nano /etc/hostapd/hostapd.conf

sudo nano /etc/default/hostapd

sudo nano /etc/sysctl.conf

sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf

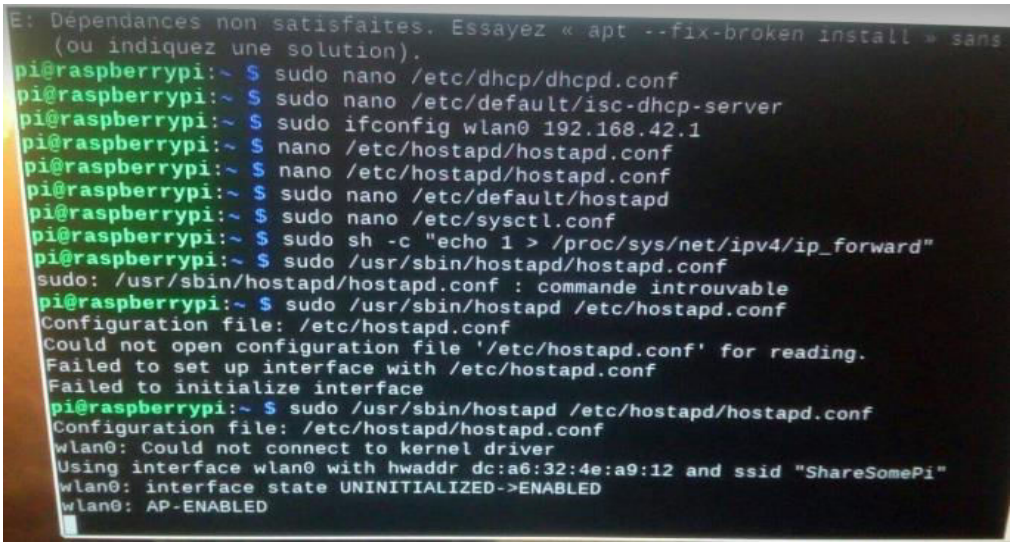


Figure 3.10: Création de réseau hotspot.

III.5.2 Configurations relatives au Raspberry pi 4

a. Installation et configuration du logiciel Samba (Figure 3.11):

sudo apt-get update

sudo apt-get install samba

sudo nano /etc/samba/smb.conf

sudo smbpasswd -a pi

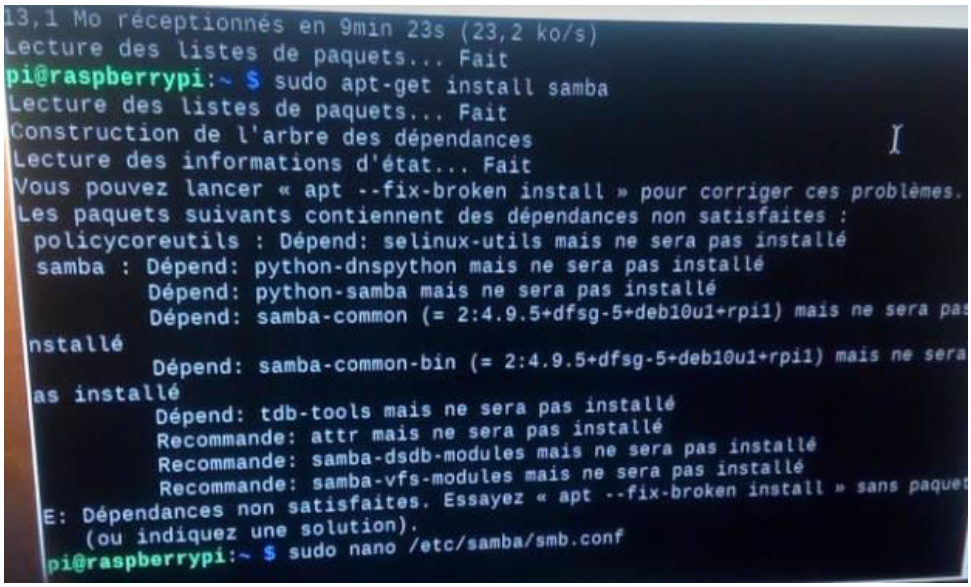


Figure 3.11: Installation et configuration du logiciel Samba.

b. Création du serveur de stockage (NAS server- Figure 3.12)

sudo mkdir /home/shares

sudo mkdir /home/shares/public

sudo chown -R root:users /home/shares/public

sudo chmod -R ug=rwx,o=rx /home/shares/public dmesg

sudo mkdir /home/shares/public/pi

sudo chown -R root:users /home/shares/public/pi

sudo chmod -R ug=rwx,o=rx /home/shares/public/pi

sudo mount /dev/sda1 /home/shares/public/pi

sudo nano /etc/fstab/dev/sda1/home/shares/public/disk1

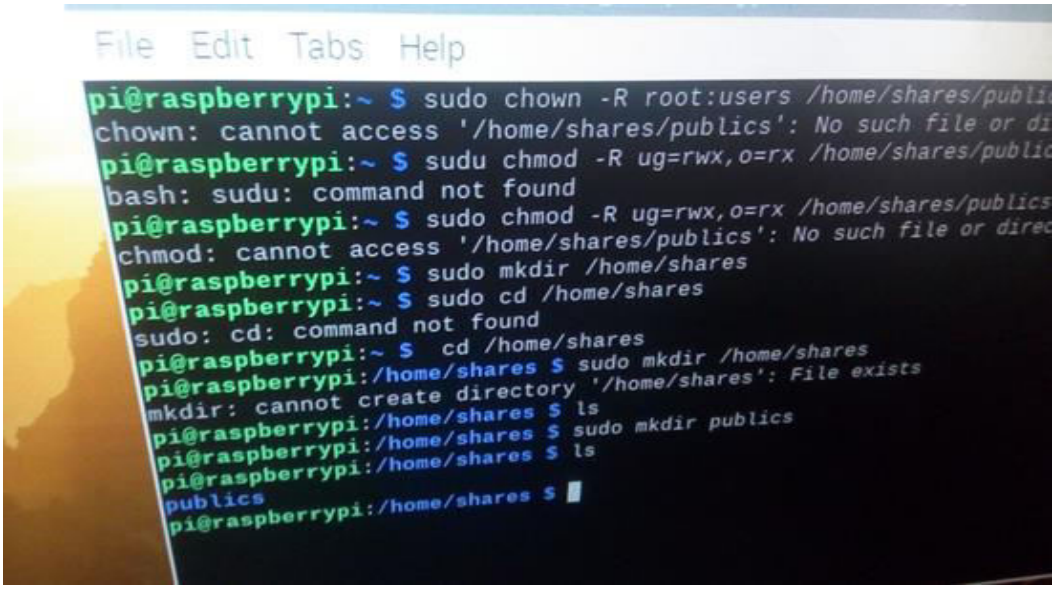


Figure 3.12: création du serveur NAS.

III.5.3 Les programmes python d'étalement et de cryptage:

a. Étalement de spectre avec le code Gold :

```
import numpy as np
import numpy.matlib
import numpy as geek
import matplotlib.pyplot as plt
from matplotlib.pyplot import step, show

import matlab.engine
eng = matlab.engine.start_matlab()

n=5
L=31

message_de_source = np.array([1,0,1,0,1,1,1,1,1,1,1,0,0,1,1])
message_source_bip=np.where(message_de_source==0, -1, message_de_source)
print("message_de_source",message_de_source)
print("message_de_source bipolaire",message_source_bip)
message_source=np.matlib.repmat(message_de_source, L, 1)
print("repmat",(message_source))

m= numpy.reshape(message_source, -1 ,order='F')
print("reshape",m)

message_bipolaire=np.where(m==0, -1, m)
print("message bipo",(message_bipolaire))

pn_code=eng.gold_ms(n,L)
print("pn code",(pn_code))
pn_code_int= np.ceil(pn_code)#convert to int
print("pn code",(pn_code_int))

pn_code_Tx=pn_code_int[0]

print("pn code tx",(pn_code_Tx))
print("len pn code",len(pn_code_Tx))
```

```
message_etale=[]
for i in (message_source_bip) :
    if i==1:
        message_etale= np.append(message_etale,pn_code_Tx)
    else:
        message_etale= np.append(message_etale,-(pn_code_Tx))

print("message etale :",(message_etale))
print("message etale :",len(message_etale))

tm = np.arange(1,len(message_de_source)*L+1,1)
li=np.arange(0,500,50)



plt.figure()
plt.subplot(3, 1, 1)
plt.title("Message de source")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-0.5, 1.5);
plt.step(tm,m)

plt.subplot(3, 1, 2)
plt.title("Message bipolaire")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-1.5, 1.5);
plt.step(tm,message_bipolaire)

plt.subplot(3, 1, 3)
plt.title("Message étalé")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-1.5, 1.5);
plt.step(tm,message_etale)
plt.show()
```

b. Étalement de spectre inverse avec le code Gold :

```
import numpy as np
import numpy.matlib
import numpy as geek
import matplotlib.pyplot as plt
from matplotlib.pyplot import step, show

import matlab.engine
eng = matlab.engine.start_matlab()

n=5
L=31

f = open("tre.enc", "r")
re=f.read()
print(re)
print(type(re))
l = [ word.strip() for word in re.split('.') ]
print(l)
print(type(l))
messag = [int(i) for i in l]

print(messag)
msg1 = numpy.array(messag)
message_etale=np.where(msg1==0, -1, msg1)


print("type etale",(message_etale))

pn_code=eng.gold_ms(n,L)
pn_code_T=pn_code[0]
 #print("pn code ",(pn_code_T))
pn_code_int= np.ceil(pn_code)#convert to int

pn_code_Tx=pn_code_int[0]
print("pn code TX",(pn_code_Tx))

longueur_message_source=int(len(message_etale)/len(pn_code_Tx));
print(len(pn_code_Tx))
```

```
longueur_message_source=int(len(message_etale)/len(pn_code_Tx));
print(len(pn_code_Tx))
message_affiche=[]
for i in range(1, longueur_message_source+1) :
        #print("i",i)

        tms = np.arange(1+(i-1)*L,(i*L)+1)
        v=tms-1
        #print("tm",v)
        #print("len tm",len(v))
        print("message_etale[v]",message_etale[v])
        #msg= [message_etale[v] * pn_code_Tx[t] for t in range(len(pn_code_Tx))]
        #msg=[a * b for a, b in zip(message_etale[v], pn_code_Tx)]
        #print("message_source",msg)
        z=(message_etale[v]*pn_code_Tx)
        print("z",z)
        message_affiche=np.append(message_affiche,z)
        print("msg",message_affiche)


#print("message_affiche",(message_affiche))
#message_affichee=message_affiche.tolist()
#msg1=np.ceil(message_affichee)
#print("message_source",msg1)
```

```
#########################################################################
message_source=[]
message_source_bip=[];
for j in range(1, longueur_message_source+1,1) :

        Y=(1+(j-1)*L)+1
        message_source_bit=message_affiche[Y+1]
        message_source_bip=np.append(message_source_bip,message_source_bit)
print("message_source_bip",(message_source_bip))
message_source_bin=np.where(message_source_bip==-1, 0, message_source_bip)
print("message_source_bin",(message_source_bin))
output = [str(int(x)) for x in message_source_bin]
print("message_source_bin",(output))
strings = [str(integer) for integer in output]
print(type(strings))
a_string = "".join(strings)
print((a_string))
an_integer = int(a_string)
print(type(an_integer))
print(an_integer)
#########################################################################
```

```
#########################################################################
srt=str(an_integer)
print(("////////////////////////////////////////////////////////////////"))
print(("////////////////////////////////////////////////////////////////"))
binary_int = int(srt, 2)
print(type(binary_int))
byte_number = binary_int.bit_length() + 7 // 8
print(type(byte_number))
binary_array = binary_int.to_bytes(byte_number, "big")
print(type(binary_array))
ascii_text = binary_array.decode()

print(ascii_text)
```

c. Cryptage et décryptage par le code César:

```
phrase = input("Ecrivez une phrase:" )
decalage = int(input("Valeur du décalage?"))
plain_text = ""

for c in phrase:


        # find the position in 0-25
        c_unicode = ord(c)

        c_index = ord(c) - ord("0")

        # perform the negative shift
        new_index = (c_index - decalage) % 75

        # convert to new character
        new_unicode = new_index + ord("0")

        new_character = chr(new_unicode)

        # append to plain string
        plain_text = plain_text + new_character


print("Encrypted text:",phrase)

print("Decrypted text:",plain_text)
```

```
phrase = input("ecrivez une phrase:")
decalage = int(input("Valeur du décalage?"))
phrase_crypte = ""

for c in phrase:

    # check if character is an uppercase letter


        # find the position in 0-25
        c_unicode = ord(c)

        c_index = ord(c) - ord("0")

        # perform the shift
        new_index = (c_index + decalage) % 75

        # convert to new character
        new_unicode = new_index + ord("0")

        new_character = chr(new_unicode)

        # append to encrypted string
        phrase_crypte = phrase_crypte + new_character


print("Plain text:",phrase)

print("Encrypted text:",phrase_crypte)
```

d. Chiffrement et déchiffrement par le code AES :

```
import hashlib
import math
import os
from Crypto import Random


from Crypto.Cipher import AES
fe = open("aes.enc", "r")
re=fe.read()
print(re.encode('cp1250'))
print(type(re))


a='aaaaaaaaaaaaaaaaaaaaaaaaaaaa'
f=a
print("phrase à cryptée ",f)

cleartext =f.encode(encoding='utf-8')
print("phrase à cryptée type utf-8 ",cleartext )
print(type(cleartext ))

iv = os.urandom(16)

key = os.urandom(16)

encrypted =  AES.new(key, AES.MODE_CFB, iv).encrypt(cleartext)


print("phrase cryptée :",encrypted)



cleartex = AES.new(key, AES.MODE_CFB, iv).decrypt(encrypted)
clearte =cleartex.decode("utf-8", "replace")
print("phrase décryptée type bytes :",cleartex)
print("phrase décryptée types utf-8 :",clearte)
```

III.6 Conclusion:

L'analyse algorithmique entamée dans ce chapitre explique en détails les organigrammes qui gèrent la génération des messages textuels numériques étalés et cryptés ; leur émission à travers le réseau conçu qui nécessite, entre autres, une configuration logicielle particulière ; et, à la fin, la réception des messages, leur étalement de spectre inverse et décryptage, Le chapitre 4 suivant présente les principaux résultats pratiques obtenus et leur discussion.

Chapitre 4 : Résultats et discussions

IV.1 Introduction

Dans ce chapitre, nous allons décrire en détails toutes les configurations relatives aux Raspberry utilisés, la carte réseau WiFi, le pont d'accès, le logiciel Samba qui génèrent les fichiers, et le serveur de stockage. D'autre part, les méthodes d'implémentation des trois techniques de sécurisation de données à savoir l'étalement de spectre, le chiffrement César et le codage AES sont présentées étape par étape.

Le langage de programmation étant le Python. Les détails relatifs au choix de bibliothèques ainsi que les instructions de cet environnement sont cités et commentés. Enfin, nous décrivons la façon dont on a adopté pour créer le logiciel telecrypt en utilisant Qt Designer.

IV.2 Transmission des fichiers

IV.2.1 Raspberry pi 1 :

a. Installation et configuration de la carte réseau WiFi :

1. Faire la mise à jour de Raspberry

```
pi@raspberry:~ $ sudo apt-get update
```

2. Exécuter pour modifier le fichier

```
pi@raspberrypi:~ $ sudo nano /etc/network/interfaces
```

3. Ajouter les informations nécessaires. Ceci inclut :

a- Configuration d'une adresse IP statique en éditant / etc / network / interfaces.
« wlan0 » étant notre carte d'interface réseau

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
iface wlan0 inet static
  address 192.168.42.1
  netmask 255.255.255.0
```

b. Création de point d'accès (hotspot) :

Dans la partie pratique, le réseau local conçu est sécurisé par WPA2. Pour créer un point d'accès, nous aurons besoin de DNSMasq et HostAPD. Il faudra installer tous les logiciels requis en une seule fois avec cette commande :

```
pi@raspberrypi:~ $ sudo apt-get install hostapd isc-dhcp-server
```

Ceci téléchargera les packages pour le point d'accès hôte.

```
pi@raspberrypi:~ $ sudo nano /etc/dhcp/dhcpd.conf
```

Ensuite, nous éditerons **/etc/dhcp/dhcpd.conf**, un fichier qui configure notre serveur DHCP. Ceci permettera aux connexions wifi d'obtenir automatiquement les adresses IP, DNS, etc.

```
#option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
```

Il faut ensuite modifier les lignes marquées par des cercles blancs pour ajouter un # au début, et donc les rendre des commentaires.. Ensuite, on doit trouver la ligne qui contient le mot « authoritative » et supprimer le # comme il est montré ci-dessous.

```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

Faire défiler vers le bas et introduire ce qui suit :

```
subnet 192.168.42.0 netmask 255.255.255.0 {
range 192.168.42.10 192.168.42.50;
option broadcast-address 192.168.42.255;
option routers 192.168.42.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8, 8.8.4.4;}
```

Cela permet d'éditer « dhcpd.conf » qui est un fichier configurant notre serveur DHCP. Cette configuration permet aux connexions WiFi d'obtenir automatiquement les adresses IP, DNS, etc. Exécuter ensuite cette ligne :

```
pi@raspberrypi:~ $ sudo nano /etc/default/isc-dhcp-server
```

```
INTERFACESv4="wlan0"
INTERFACESv6=""
```

Pour donner un nom à la carte réseau que nous voulons utiliser, il faut défiler jusqu'à INTERFACES = "" et les mettre soit, par exemple, INTERFACES = "wlan0".

Nous pouvons maintenant configurer les détails du point d'accès. Nous allons mettre en place un réseau protégé par un mot de passe afin que seules les personnes disposant de ce dernier puissent se connecter.

```
pi@raspberrypi:~ $ sudo nano /etc/hostapd/hostapd.conf
```

```
interface=wlan0
driver=nl80211
ssid=Pi_AP
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=miramira
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Nous allons maintenant indiquer au Rasbperry Pi où trouver ce fichier de configuration.

```
pi@raspberry:~ $ sudo nano /etc/default/hostapd
```

A chaque fois qu'on lance le hotspot, la lecture de fichier de configuration de notre réseau (hostapd.conf) sera automatique.

```
DAEMON_CONF="etc/hostapd/hostapd.conf"
```

La configuration de NAS permettra à plusieurs clients de se connecter au WiFi et d'avoir toutes les données «tunnelisées» via la seule adresse IP Ethernet. (Il faut le faire même si un seul client va se connecter).

```
pi@raspberry:~ $ sudo nano /etc/sysctl.conf
```

Faire défiler vers le bas et ajouter :

```
#kernel.sysrq=438

net.ipv4.ip_forward=1
```

Finalement :

```
pi@raspberrypi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

Si on exécute cette dernière commande, le point d'accès est activé:

```
pi@raspberrypi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr dc:a6:32:4e:a9:12 and ssid "ShareSomePi"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

IV.2.2 Raspberry Pi 4 :

a. Logiciel Samba :

Nous avons créé un serveur de stockage en réseau, également appelé stockage en réseau NAS (Network Attached Storage). Avec cette technique, la gestion des sauvegardes des données d'un réseau devient plus facile. Nous utilisons des disques de grande capacité (nous avons utilisé un flash disque pour le test). Le serveur permettra l'accès via plusieurs postes clients aux mêmes données stockées sur le NAS et aussi la réduction du temps d'administration des postes clients en gestion d'espace disque.

Nous allons créer les dossiers publics et privés qui seront accessibles sur le NAS.

```
pi@raspberrypi:~ $ sudo mkdir /home/shares
pi@raspberrypi:~ $ sudo mkdir /home/shares/public
pi@raspberrypi:~ $ sudo chown -R root:users /home/shares/public
pi@raspberrypi:~ $ sudo chmod -R ug=rwx,o=rx /home/shares/public
pi@raspberrypi:~ $
```

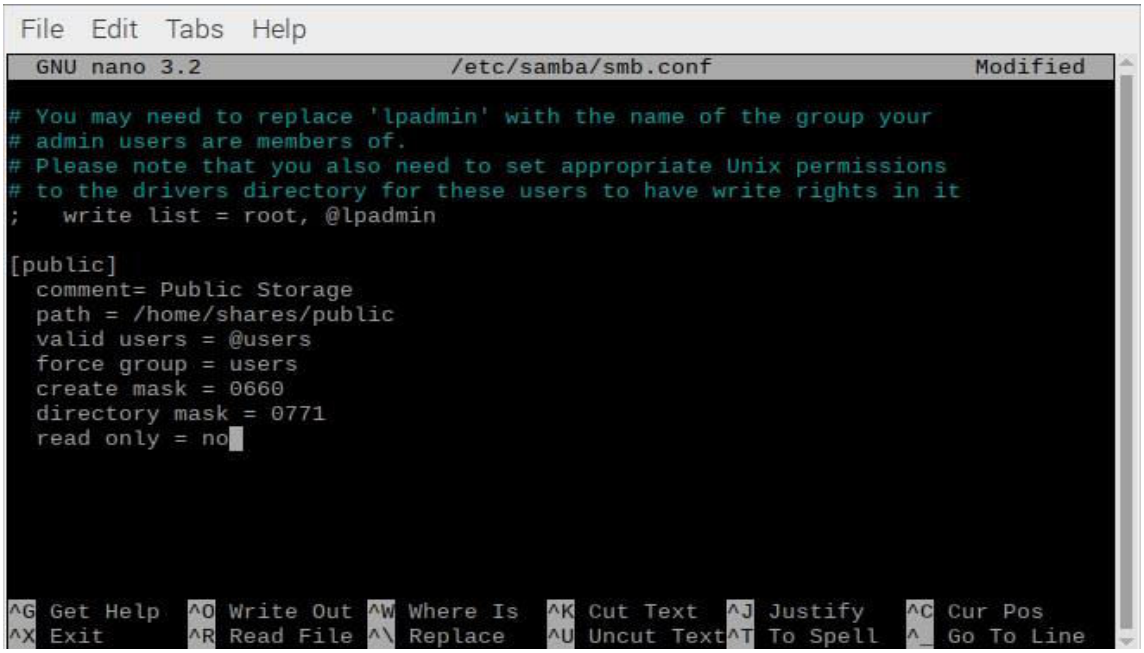
Installation de Samba :

```
pi@raspberrypi:~ $ sudo apt-get install samba
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Vous pouvez lancer « apt --fix-broken install » pour corriger ces problèmes.
Les paquets suivants contiennent des dépendances non satisfaites :
 policycoreutils : Dépend: selinux-utils mais ne sera pas installé
 samba : Dépend: python-dnspython mais ne sera pas installé
         Dépend: python-samba mais ne sera pas installé
         Dépend: samba-common (= 2:4.9.5+dfsg-5+deb10u1+rpi1) mais ne sera pas
nstallé
         Dépend: samba-common-bin (= 2:4.9.5+dfsg-5+deb10u1+rpi1) mais ne sera
as installé
```

Configuration de Samba :

```
pi@raspberrypi:~ $ sudo nano /etc/samba/smb.conf
```

Tout en bas du fichier, nous allons rajouter des paramètres relatifs à l'accès à la partie public du NAS (plusieurs parties publiques et privées peuvent être crées) :



b. Serveur de stockage (NAS server) :

A présent, nous allons ajouter un utilisateur à Samba. Dans notre exemple nous ajouterons l'utilisateur pi.

```
pi@raspberrypi:~ $ sudo smbpasswd -a pi
```

Il faudra commencer par brancher notre périphérique (flash disque USB) à notre Raspberry pi, créer ensuite un répertoire dans lequel sera monté le périphérique pour qu'il soit accessible via le NAS. Il faudra lui donner les droits de lecture/écriture nécessaires.

```
pi@raspberrypi:~ $ sudo mkdir /home/shares/public/pi
pi@raspberrypi:~ $ sudo chown -R root:users /home/shares/public/pi
pi@raspberrypi:~ $ sudo chmod -R ug=rwx,o=rx /home/shares/public/pi
pi@raspberrypi:~ $
```

Une fois fait, on exécute la commande suivante :

```
sudo mount /dev/sdb1 /home/shares/public/pi
```

Notre NAS est maintenant configuré. Pour les Smartphones, ils peuvent se connecter au NAS créé en utilisant une application adéquate comme File Expert pour Android ou File Explorer sur IOS

Résultat sur système d'exploitation Android :

On ouvre l'application File Expert (Figure 4.1) pour permettre une connexion NAS.

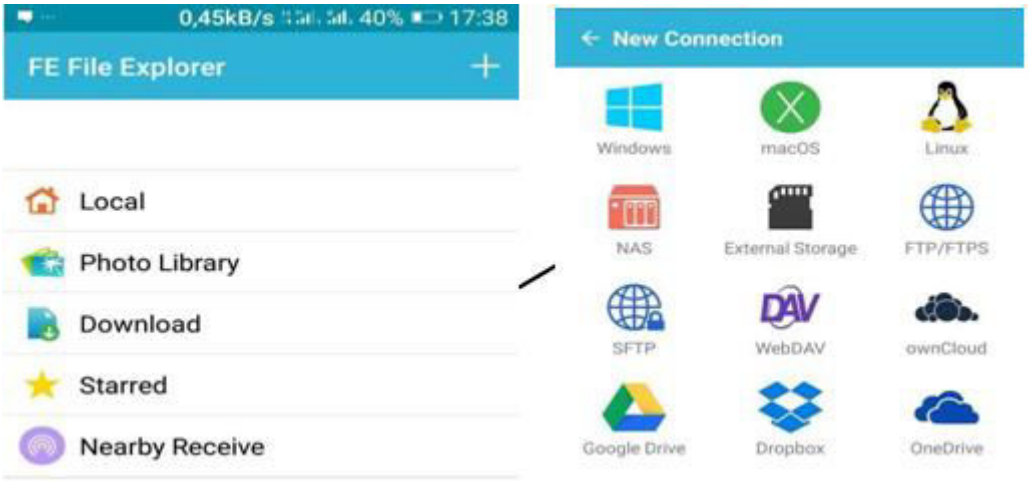


Figure 4.1 *:* Application File Expert.

Une fois l'application lancée, on introduit l'adresse IP de notre Raspberry, nom de notre utilisateur pi et le mot de passe du fichier partagé (Figure 4.2).

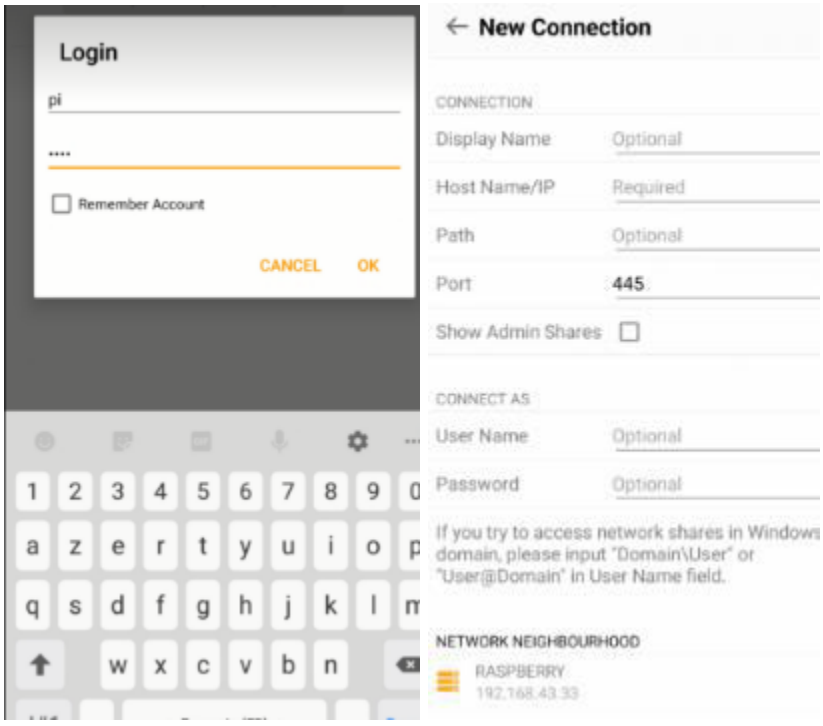


Figure 4.2*:* Configuration d'une connexion sur un appareil Android.

La Figure 4.3 montre les fichiers de flash disque partagés dans le réseau en particulier le dossier pi.

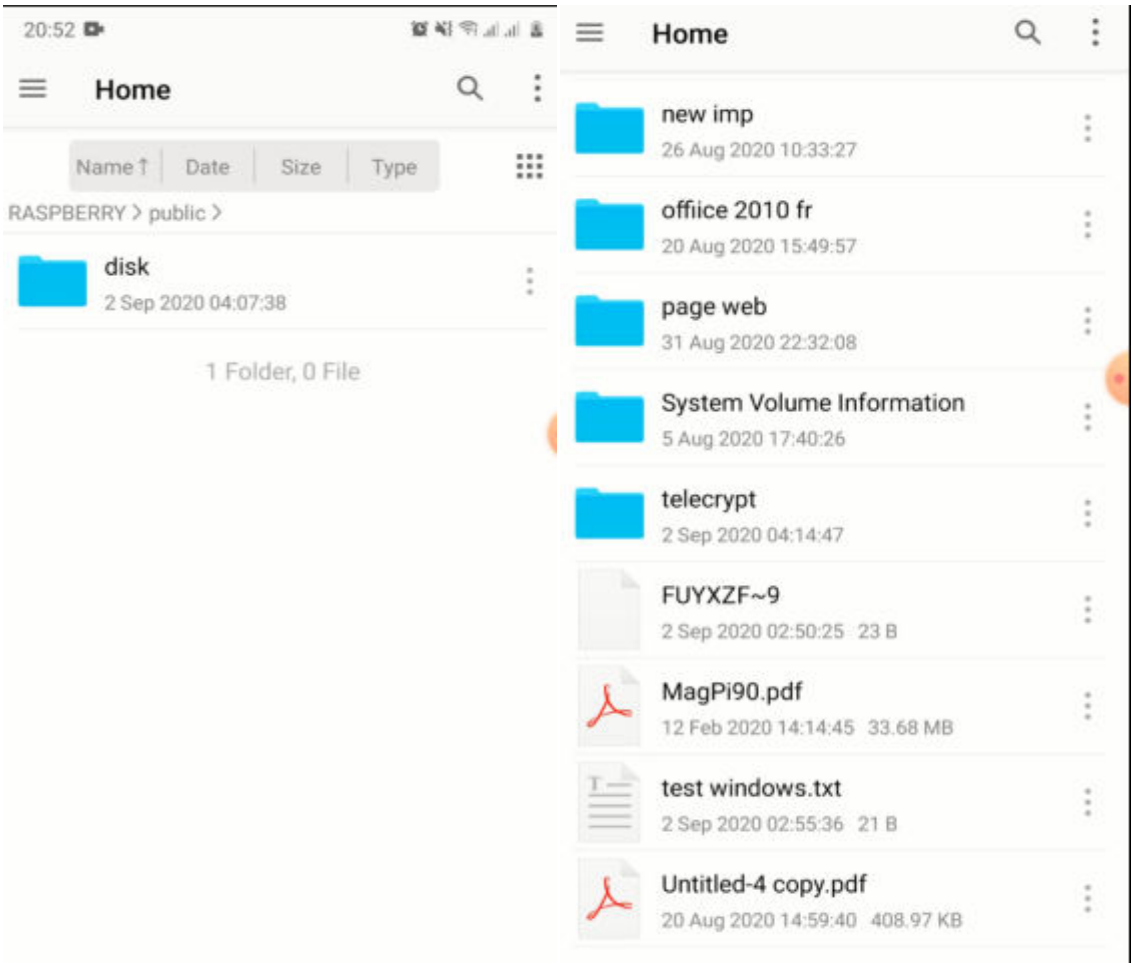


Figure 4.3: Les fichiers de flash disque.

Résultats sur système d'exploitation Windows :

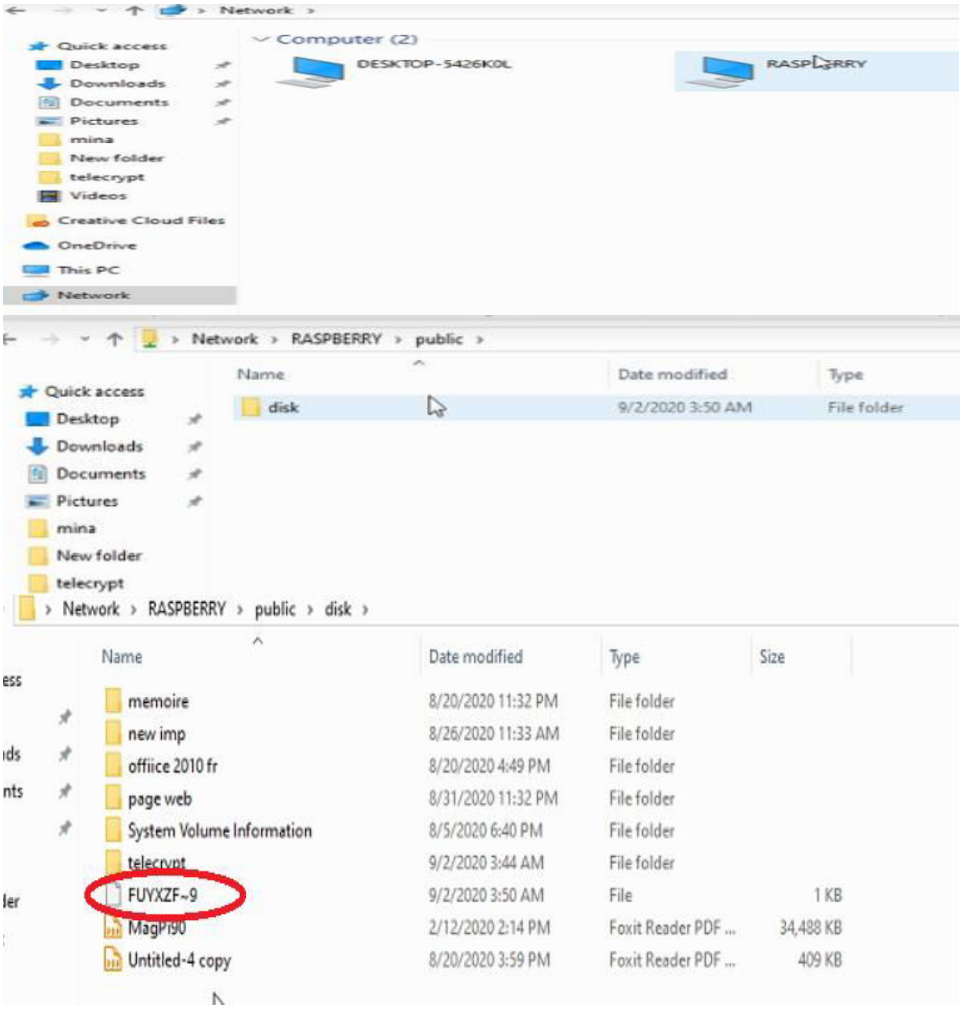


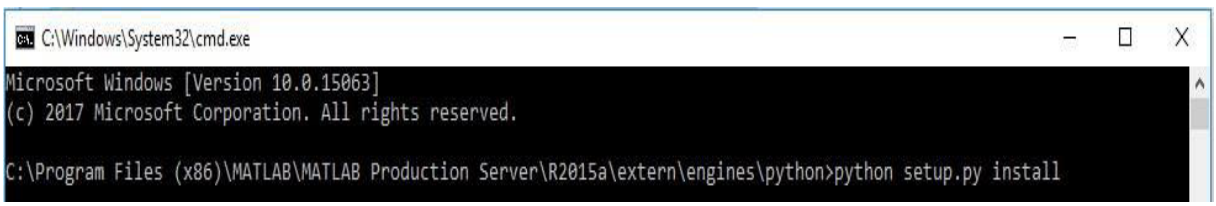
Figure 4.3: Les fichiers de flash disque.

IV.3 Sécurisation de données

A ce stade nous avons utilisé deux méthodes de cryptographie à savoir le code de César et le cryptage par AES précédées par étalement du spectre avec le code Gold.

IV.3 .1 Étalement du spectre avec code Gold :

1) Installation de bibliothèque **engine:**



L'installation de bibliothèque **engine** étant réussit.

```
copying dist\matlab\mlexceptions.py -> build\lib\matlab
copying dist\matlab\__init__.py -> build\lib\matlab
creating build\lib\matlab\engine
copying dist\matlab\engine\engineerror.py -> build\lib\matlab\engine
copying dist\matlab\engine\enginesession.py -> build\lib\matlab\engine
copying dist\matlab\engine\futureresult.py -> build\lib\matlab\engine
copying dist\matlab\engine\matlabengine.py -> build\lib\matlab\engine
copying dist\matlab\engine\__init__.py -> build\lib\matlab\engine
creating build\lib\matlab\_internal
copying dist\matlab\_internal\mlarray_sequence.py -> build\lib\matlab\_internal
copying dist\matlab\_internal\mlarray_utils.py -> build\lib\matlab\_internal
copying dist\matlab\_internal\__init__.py -> build\lib\matlab\_internal
running install_lib
creating C:\Python34\Lib\site-packages\matlab
creating C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\engineerror.py -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\enginesession.py -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\futureresult.py -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\matlabengine.py -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\_arch.txt -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\engine\__init__.py -> C:\Python34\Lib\site-packages\matlab\engine
copying build\lib\matlab\mlarray.py -> C:\Python34\Lib\site-packages\matlab
copying build\lib\matlab\mlexceptions.py -> C:\Python34\Lib\site-packages\matlab
creating C:\Python34\Lib\site-packages\matlab\_internal
copying build\lib\matlab\_internal\mlarray_sequence.py -> C:\Python34\Lib\site-packages\matlab\_internal
copying build\lib\matlab\_internal\mlarray_utils.py -> C:\Python34\Lib\site-packages\matlab\_internal
copying build\lib\matlab\_internal\__init__.py -> C:\Python34\Lib\site-packages\matlab\_internal
copying build\lib\matlab\__init__.py -> C:\Python34\Lib\site-packages\matlab
byte-compiling C:\Python34\Lib\site-packages\matlab\engine\engineerror.py to engineerror.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\engine\enginesession.py to enginesession.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\engine\futureresult.py to futureresult.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\engine\matlabengine.py to matlabengine.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\engine\__init__.py to __init__.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\mlarray.py to mlarray.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\mlexceptions.py to mlexceptions.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\_internal\mlarray_sequence.py to mlarray_sequence.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\_internal\mlarray_utils.py to mlarray_utils.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\_internal\__init__.py to __init__.cpython-34.pyc
byte-compiling C:\Python34\Lib\site-packages\matlab\__init__.py to __init__.cpython-34.pyc
running install_egg_info
Writing C:\Python34\Lib\site-packages\matlabengineforpython-R2015a-py3.4.egg-info

C:\Program Files (x86)\MATLAB\MATLAB Production Server\R2015a\extern\engines\python>
```

2) Maintenant on ouvre l'éditeur Python pour créer un nouveau fichier (Figure 4.4) :

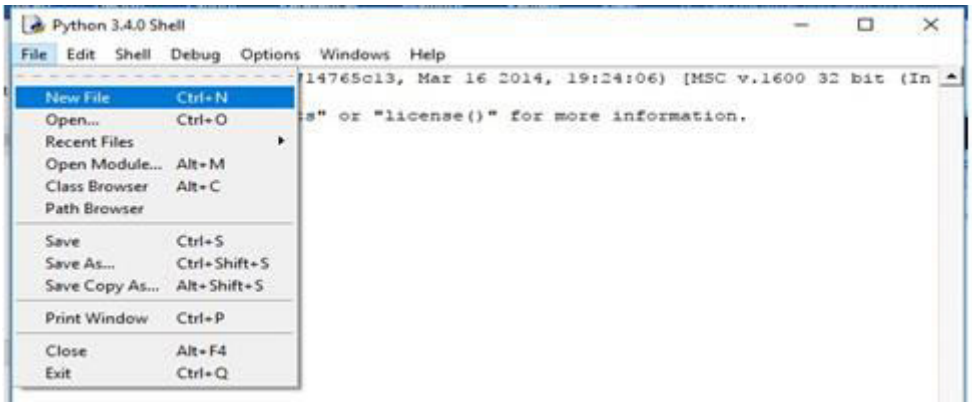


Figure 4.4: nouvelle fichier python.

3) On appelle les bibliothèques requises suivantes:

- La bibliothèque **NumPy** permet d'effectuer des calculs numériques avec Python.

Elle introduit une gestion facilitée des tableaux de nombres.

```
import numpy as np
import numpy.matlib
import numpy as geek
```

- Ensuite on fait appeler la bibliothèque **Matplotlib** pour permettre de générer directement des graphiques à partir de Python. Au fil des années, Matplotlib est devenue une librairie puissante, compatible avec beaucoup de plateformes, et capable de générer des graphiques dans beaucoup de formats différents.

```
import matplotlib.pyplot as plt
from matplotlib.pyplot import step, show
```

- Pour permettre la génération du code Gold développé sur Matlab et l'utiliser dans l'environnement Python, on doit se servir de la bibliothèque **MATLAB. Engine**. Ceci fournit un package permettant à Python d'appeler Matlab en tant que moteur de calcul. Le moteur prend en charge l'implémentation de référence (CPython).

La commande ci-dessous permet de démarrer **MATLAB.Engine** pour Python.

```
import matlab.engine
eng = matlab.engine.start_matlab()
```

4) Génération de la séquence étalée par le code Gold

Nous déclarons tout d'abord le nombre de registres à décalage n et donc la longueur du code Gold utilisé $L=2^n-1$. Après on génère un message de source binaire contenu dans un vecteur nommé np.array.

```
n=5
L=31
message_de_source = np.array([1,0,1,0,1,1,1,1,1,1,1,0,0,1,1])
print("message_de_source",message_de_source)
```

Ensuite, on répète notre message L fois en accord avec la longueur du code PN utilisé.

```
message_source=np.matlib.repmat(message_de_source, L, 1)
print("repmat",(message_source))
```

La matrice après cette opération, appelée « message_source » s'affiche comme suit :

1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1

1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	1	1	1	1	0	0	1	1

Ensuite, on convertit la matrice « message_source » en un vecteur ligne en utilisant la commande ci-après suivie d'un affichage.

```
message_de_source= numpy.reshape(message_source, -1 ,order='F')
print("reshape", message_de_source)
```

Le résultat s'affiche comme le montre la Figure 4.5.

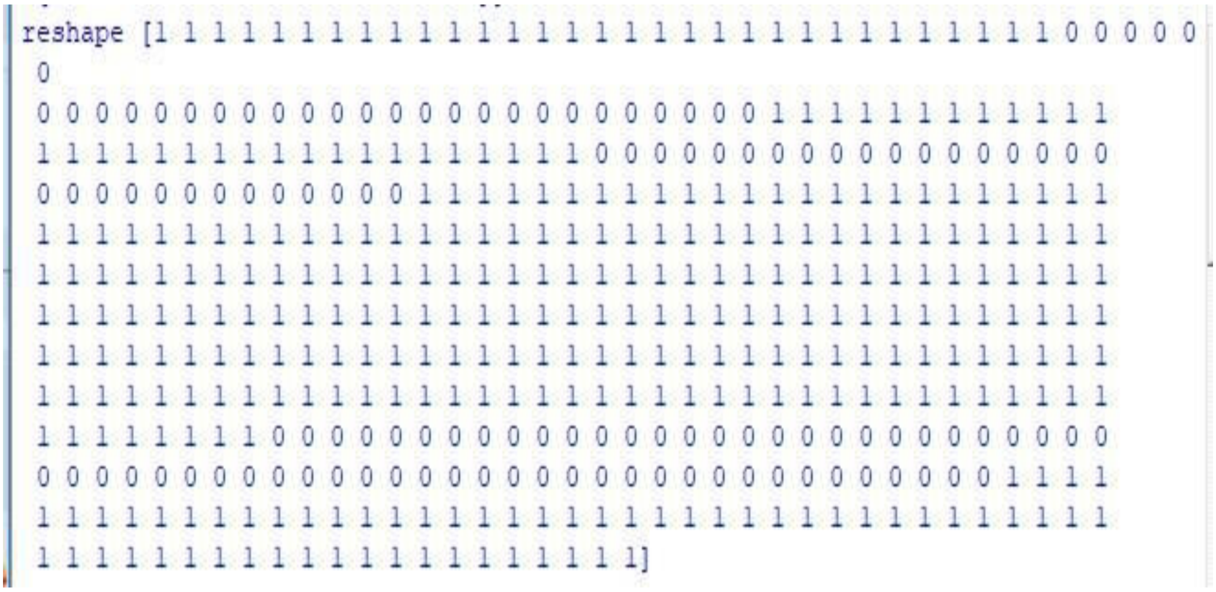

```
reshape [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

Figure 4.5 : Résultat sur Python de la commande reshape appliquée sur la matrice « message_source ».

Pour convertir du binaire 0/1 vers le bipolaire -1/1, on procède comme suit :

```
message_bipolaire=np.where(message_de_source==0, -1, message_de_source)
print("message bip :",(message_bipolaire))
```

Le résultat de bipolarisation s'affiche sur la Figure 4.6.

```
message bip [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1
  1  1  1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1]
```

Figure 4.6 : Résultat de conversion du message binaire en message bipolaire -1/1.

Nous avons besoin du code pseudo-aléatoire de type Gold qui est déjà développé sous Matlab. On le fait appeler puis l'afficher grace aux instructions suivantes :

```
pn_code=eng.gold_ms(n,L)
print("pn code",(pn_code))
```

Le résultat obtenu est reporté sur La Figure 4.7. Après on convertit le code PN en format entier (voir Figure 4.8).

```
y = np.ceil(pn_code)
print("pn code",(y))
pn_code_Tx=pn_code[0]
```

Enfin, on choisit la première séquence du code Gold parmi les *L* séquences possibles en procédant comme suit :

```
pn_code_Tx=pn_code[0]

print("pn code tx",(z))
```

La séquence utilisée dans l'étalement est affichée sur la Figure 4.9

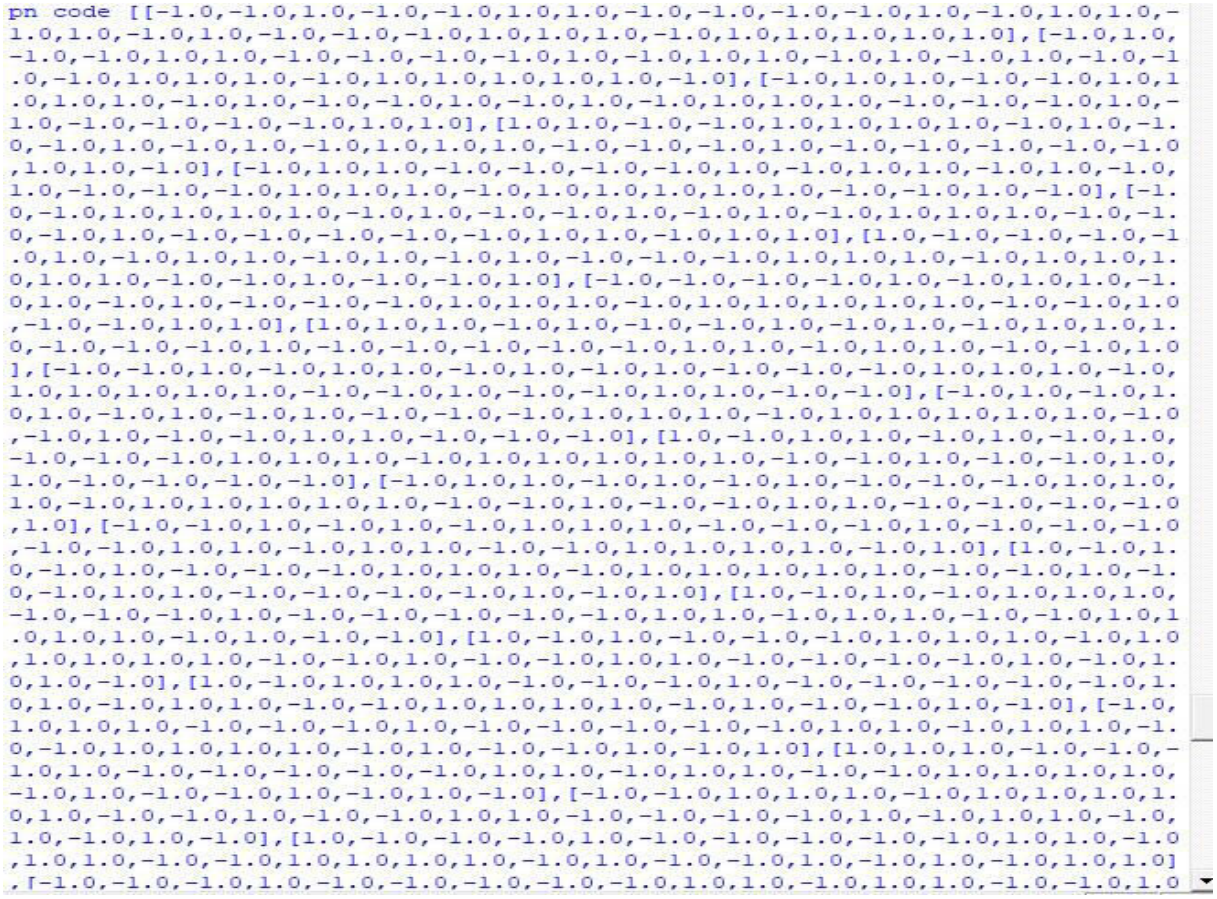


Figure 4.7: Résultat obtenu sur Ppython relatif au code Gold. Chaque ligne est un code PN de type Gold.

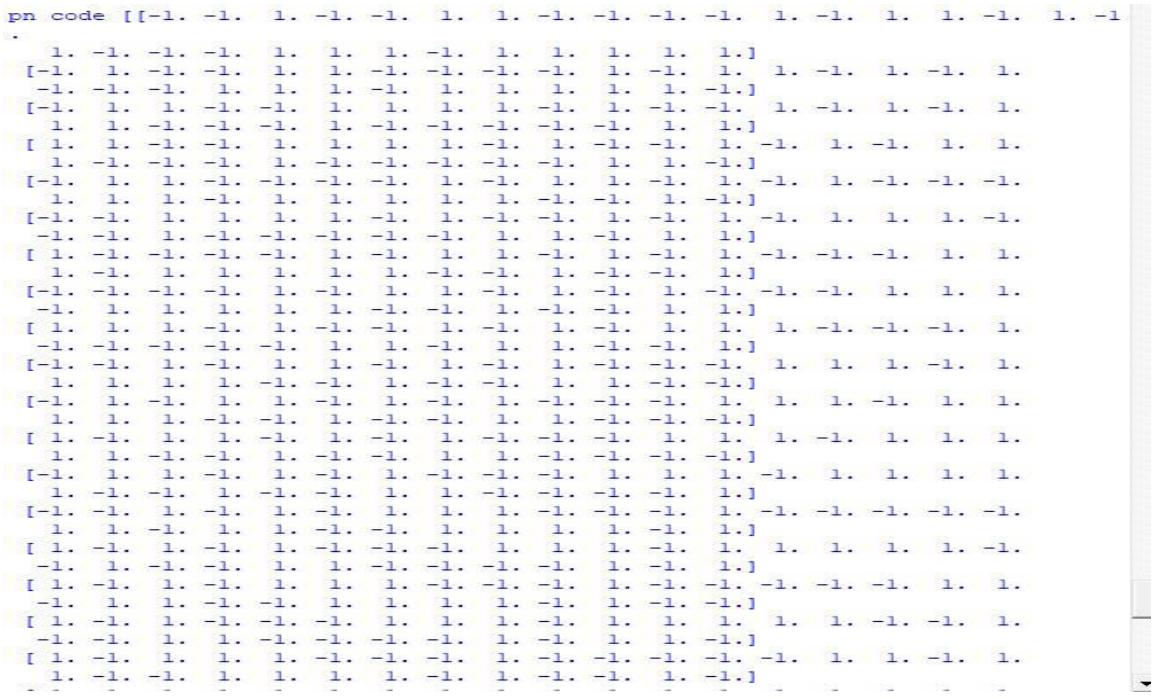


Figure 4.8: Les 31 séquences du code Gold après sa conversion en format entier.

Figure 4.9: La 1ère ligne de la matrice de la Figure 4.8 prise autant que séquence Gold.

Il reste juste à générer le message étalé et afficher les trois graphes en suivant les instructions ci-après :

```
#Etalement de spectre
message_etale=[]
for i in (message_bipolaire) :
    if i==1:
        message_etale= np.append(message_etale,z)
    else:
        message_etale= np.append(message_etale,-(z))
print("message etale :",(message_etale))


#Tracé des graphes
plt.figure()
plt.subplot(3, 1, 1)
plt.title("Message de source")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-0.5, 1.5);
plt.step(tm,m)
plt.subplot(3, 1, 2)
plt.title("Message bipolaire")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-1.5, 1.5);
plt.step(tm,c)
plt.subplot(3, 1, 3)
plt.title("Message étalé")
plt.grid(color='b', ls = '-.', lw = 0.25)
plt.xticks(np.arange(0,500,25))
plt.ylim(-1.5, 1.5);
plt.step(tm,message_etale)
plt.show()
```

Les résultats de l'étalement sous Python sont affichés sur la Figure 4.10.

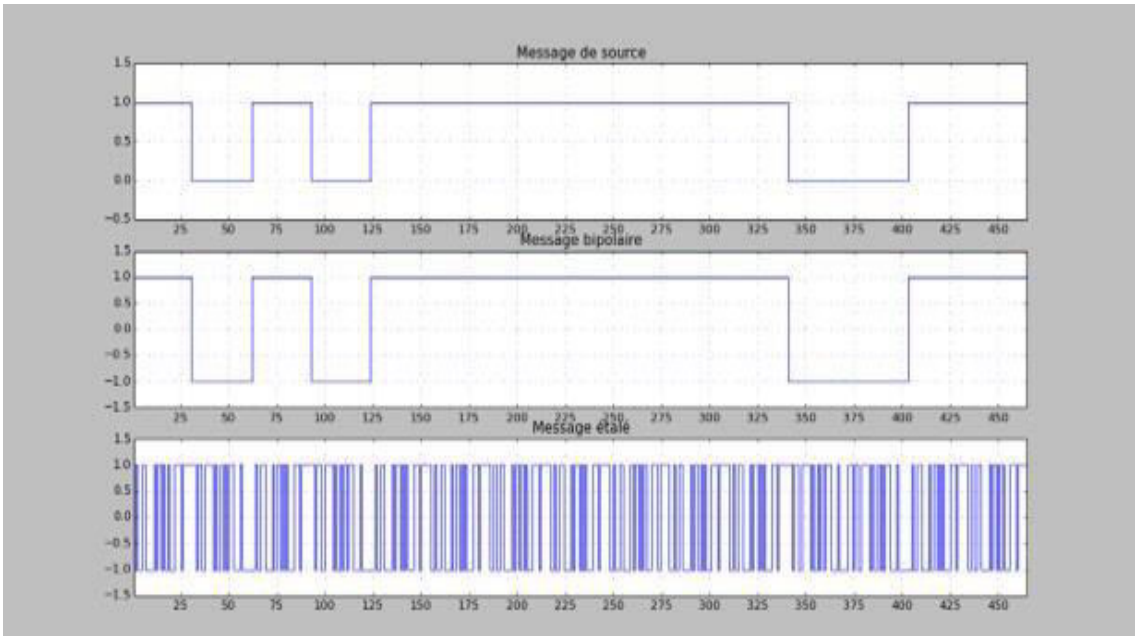


Figure 4.10: Affichage sous Python du message de source, message de source bipolaire et du message étalé à séquence directe.

IV.3.2 Le code César :

Le code César (ou chiffre de César) est un chiffrement par décalage parmi les plus connus, il utilise la substitution d'une lettre par une autre plus loin dans l'alphabet et les chiffres.

a.Chiffrement César :

```
phrase = input("Ecrivez une phrase:" )
decalage = int(input("Valeur du décalage?"))
phrase_crypte = ""
```

Déclaration des entrées (phrase, décalage) et des sorties (phrase_crypte).

```
Ecrivez une phrase:telecom
Valeur du décalage? 2
```

L'utilisateur attribut une valeur de décalage et la phrase à crypter

```
for c in phrase:
```

Cette boucle for permet de lire à chaque fois les lettres de la phrase à crypter (lettre par lettre)

```
c : t
```

La première lettre dans la phrase est : t

```
    c_unicode = ord(c)
    c_index = ord(c) - ord("0")
```

c_unicode reçoit l'ordre de notre lettre lire (c_unicode est un nombre int).

c_index reçoit la soustraction de l'ordre de notre lettre et l'ordre du premier caractère que nous allons choisi (zéro dans notre exemple). Le but est de déterminer le nombre de caractères entre le premier caractère et le caractère présent.

```
c_unicode: 116
c_index: 68
```

new_index = (c_index + decalage) % 75

On ajoute le numéro de décalage au c_index

```
new_index: 70
```

new_unicode = new_index + ord("0")

(%75) 75 c'est le nombre de caractères que nous avons choisi : les chiffre de 0 à 9 et les lettres de a à z et de A à Z.

Pour avoir l'ordre du caractère on ajoute l'ordre du premier caractère.

```
new_unicode: 118
```

new_character = chr(new_unicode)

On utilise la commande chr pour connaitre le caractère de l'ordre (new_unicode)

```
new_character: v
```

Donc le décalage de t par 2 c'est le v.

phrase_crypte = phrase_crypte + new_character

On sauvegarde le caractère dans phrase_crypte

```
phrase_crypte: v
```

Ces étapes sont répétées dans la boucle for.

print("Plain text:",phrase)

print("Encrypted text:",phrase_crypte)

A la fin, le résultat de cryptage s'affiche comme suit :

```
phrase_crypte: vgngeqo
Plain text: telecom
```

b.Déchiffrement César:

phrase = input("Ecrivez une phrase:")

decalage = int(input("Valeur du décalage?"))

plain_text = ""

Déclaration des entrées (phrase, décalage) et de sortie (plain_text)

```
Ecrivez une phrase:vgngeqo
Valeur du décalage?2
```

L'utilisateur attribut une valeur de décalage et la phrase à crypter :

for c in phrase

Cette boucle sert à lire la phrase cryptée caractère par caractère.

```
c : v
```

La première lettre dans la phrase est :

c_unicode reç c_unicode = ord(c)

c_index = ord(c) - ord("0")

De même qu'au préalable, on obtient :

```
c_unicode: 118
c_index: 70
```

new_index = (c_index - decalage) % 75

On soustrait c_index et le numéro de décalage

```
new_index: 68
```

new_unicode = new_index + ord("0")

Pour voir l'ordre du caractère on ajoute l'ordre de premier caractère.

```
new_unicode: 116
```

new_character = chr(new_unicode)

On utilise la commande chr pour connaitre le caractère de l'ordre (new_unicode)

```
new_character: t
```

Donc le décalage par 2 en arrière de v donne t

plain_text = plain_text + new_character

On sauvegarde le caractère dans plain_text

```
plain_text: t
```

Ces étapes sont répétées dans la boucle for

print("Encrypted text:",phrase)

print("Decrypted text:",plain_text)

Après ces étapes, on obtient la phrase décryptée qui correspond parfaitement au message de la source :

```
Encrypted text: vgngeqo
Decrypted text: telecom
```

IV.3.3 Le code AES :

a.Chiffrement AES

L'appel des bibliothèques :

```
import hashlib
import math
import os
a='aaaaaaaaaaaaaaaaaaaaaaaaaa'
f=a
print(f)
cleartext =f.encode(encoding='utf-8')
iv = os.urandom(16)
key = os.urandom(16)
encrypted =  AES.new(key, AES.MODE_CFB, iv).encrypt(cleartext)
```

f correspond au message de la source qui sera crypter en utilisant la commande (f. encode), (utf-8 : veut dire en anglais Universal caracter set transformation format-8 bits), cleartext reçoit la conversion de f en type bytes.

```
phrase à cryptée  aaaaaaaaaaaaaaaaaaaaaaaaaa
phrase à cryptée type utf-8  b'aaaaaaaaaaaaaaaaaaaaaaaaaa'
```

Le code AES a deux paramètres importants le iv et la clé.

Dans notre exemple nous avons choisi une clé et un iv aléatoires de taille 16 bit.

Ceci fait, on choisit le mode de cryptage dans notre exemple c'est mode CFB. La commande encrypt génère la phrase cryptée comme suit :

```
encrypted =  AES.new(key, AES.MODE_CFB, iv).encrypt(cleartext)
print("code :",encrypted)
```

Encrypted reçoit la phrase cryptée :

```
phrase cryptée : b's\xa2\xbe>EV\x1f>\xa5\xaa\xde/\x8f\xd9\xa6{V\xfd\xe9\x9e\xa99\x91\x16\x8a\xa4\xcd'
```

b. Déchiffrement AES :

Pour décrypter la phrase on garde les mêmes paramètres (clé et iv) et le même mode (CFB)

```
cleartex = AES.new(a, AES.MODE_CFB, iv).decrypt(encrypted)
clearte =cleartex.decode("utf-8", "replace")
```

```
print("phrase décryptée type bytes : " ,cleartex)
print("phrase décryptée type utf-8 : " , clearte)
```

On utilise la commande decrypt pour décrypter la phrase. La phrase s'affiche de type bytes.

```
phrase décryptée type bytes : b'aaaaaaaaaaaaaaaaaaaaaaaaaa'
```

Maintenant on convertit la phrase de types bytes en type string:

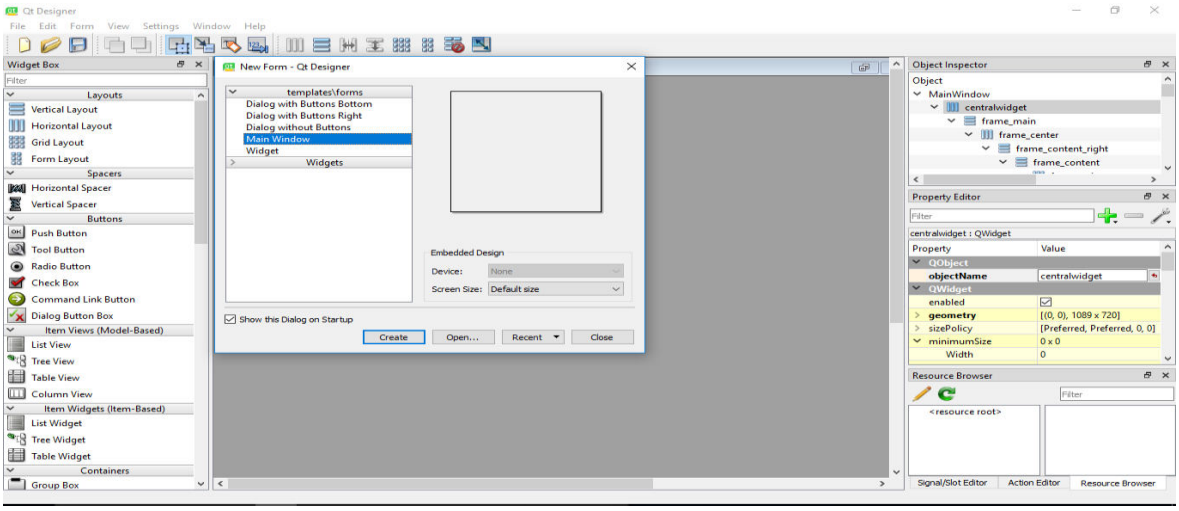
```
phrase décryptée types utf-8 : aaaaaaaaaaaaaaaaaaaaaaaaaa
```

IV.4 Logiciels telecrypt :

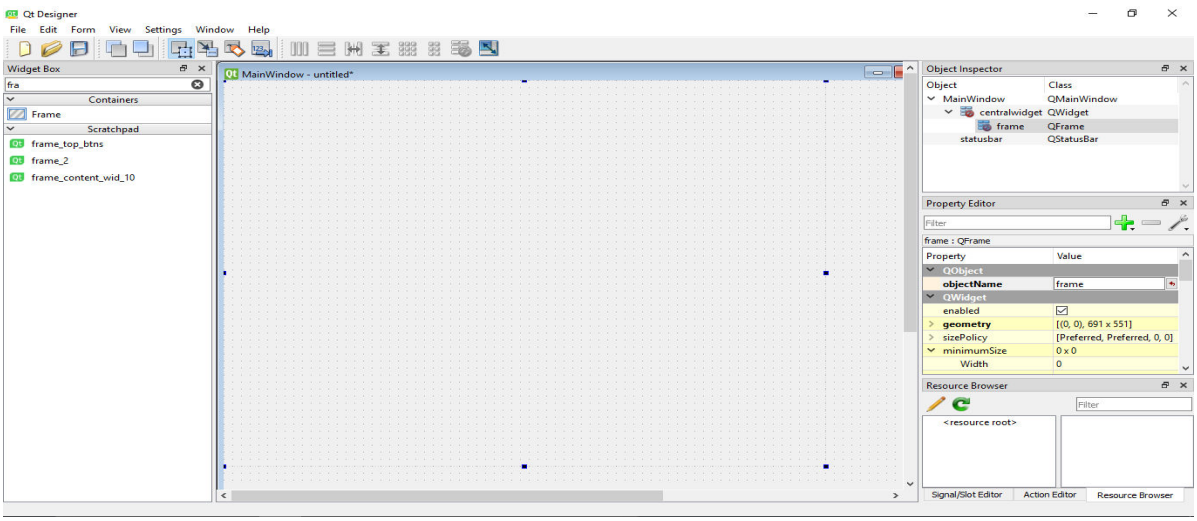
Logiciel Qt Designer:

On ouvre le logiciel Qt Designer. La création de l'interface graphique est expliquée ci-dessous.

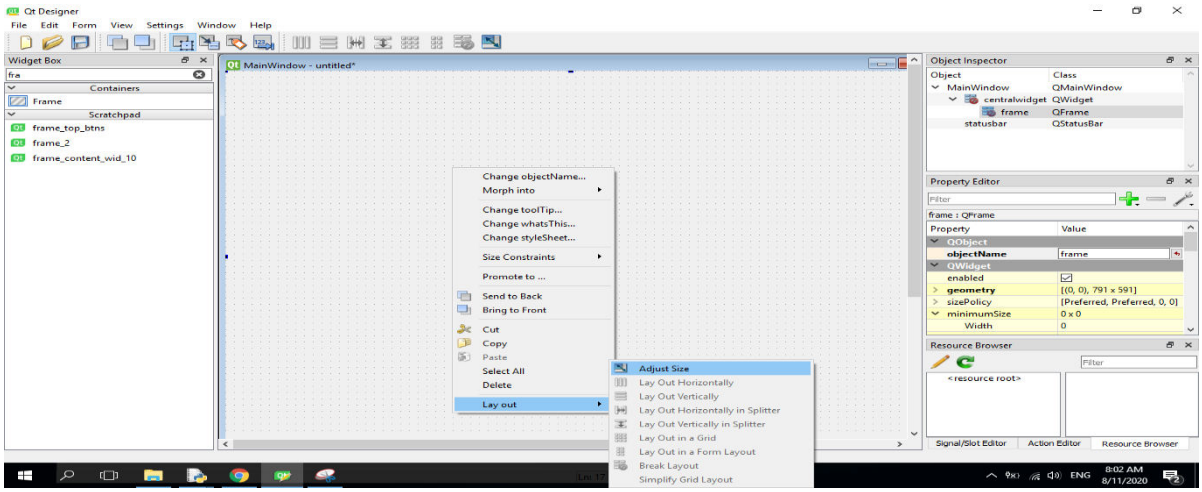
- On commence par créer une fenêtre (main window).



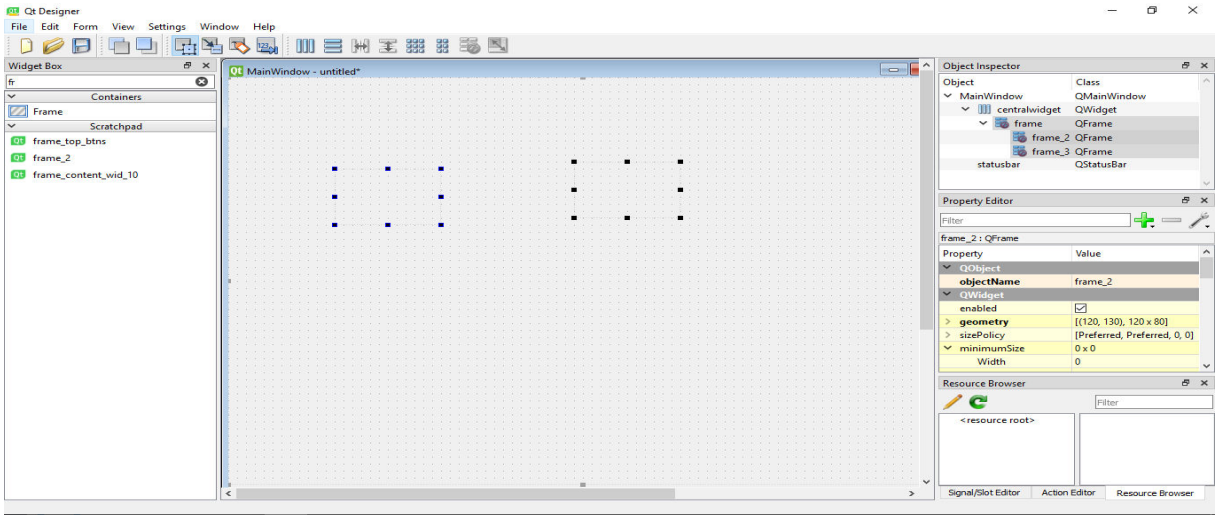
-On ajoute un frame (frame principle).



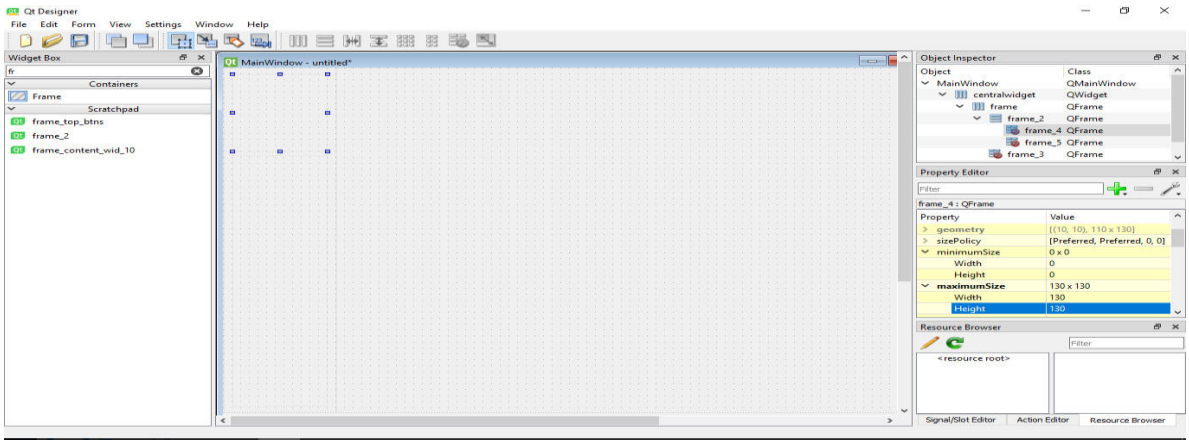
-On ajuste la taille de frame



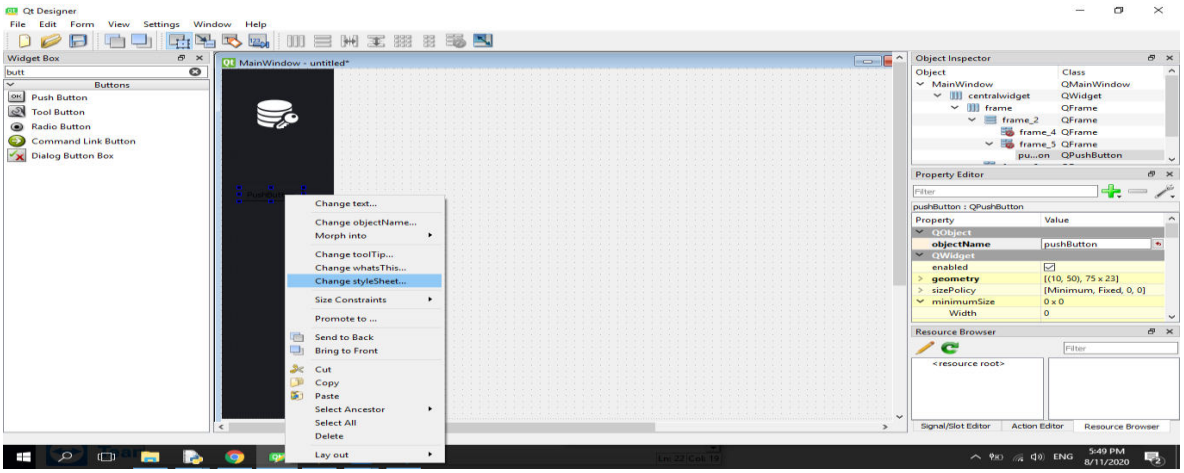
-On créé un frame pour menu et une frame pour le contenu:



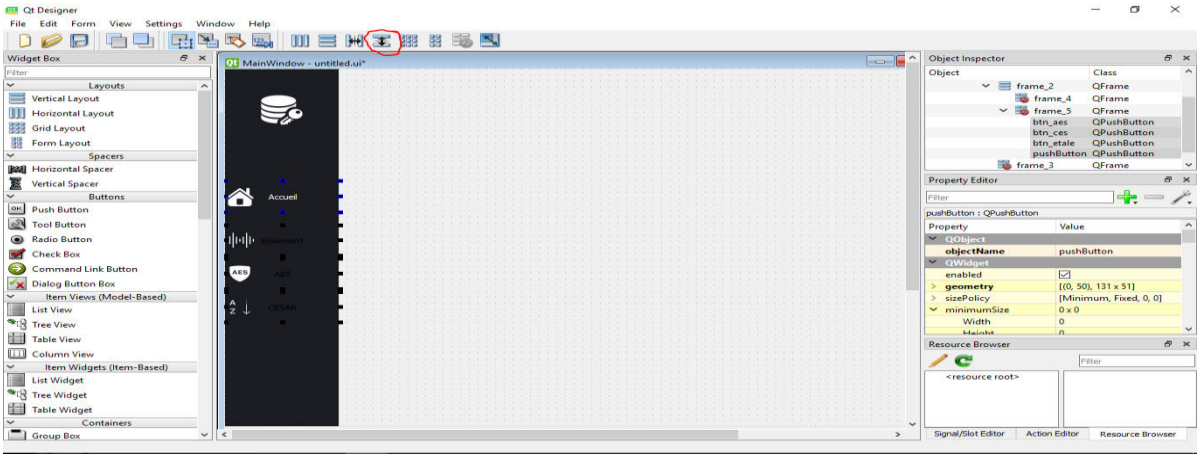
-On ajoute un frame pour le logo de notre logiciel :



-Maintenant, nous avons besoin de créer des Boutons:



-Pour séparer les boutons on clique sur la commande entourée par le cercle rouge :



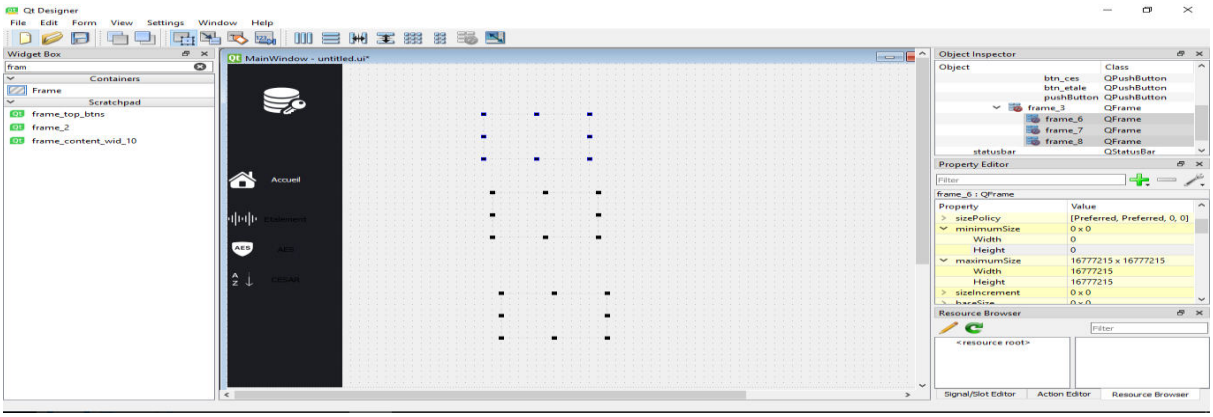
Le programme permettant le paramétrage d'un bouton est introduit ci-dessous :

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1089, 720)
        MainWindow.setMinimumSize(QtCore.QSize(1000, 720))
        palette = QtGui.QPalette()
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
```

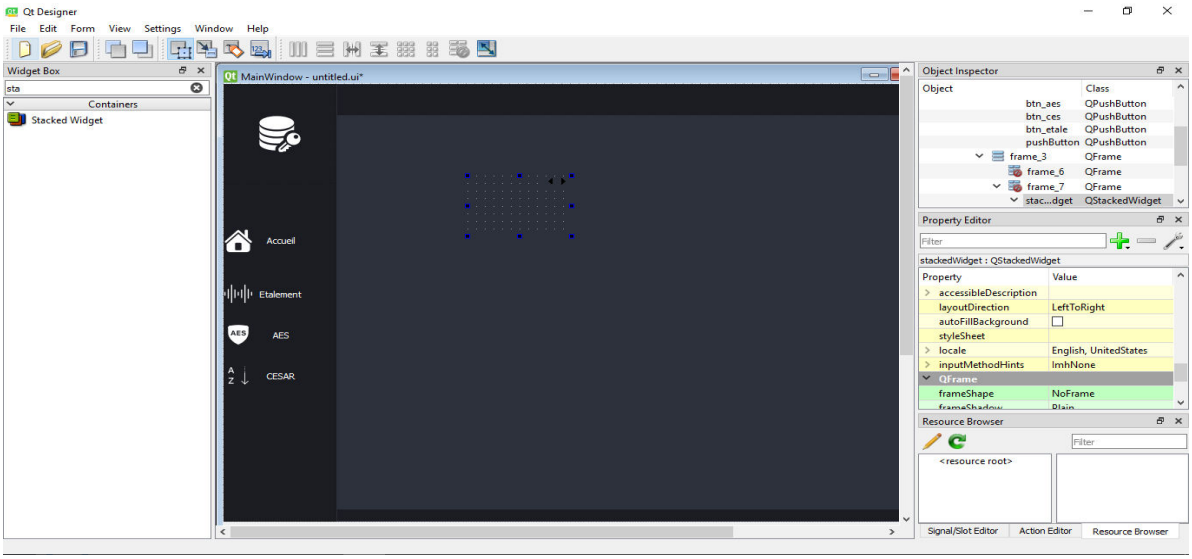
```
brush = QtGui.QBrush(QtGui.QColor(66, 73, 90))
brush.setStyle(QtCore.Qt.SolidPattern)
```

La barre de menu est maintenant accomplie.

-Maintenant on créé trois frames placés verticalement :

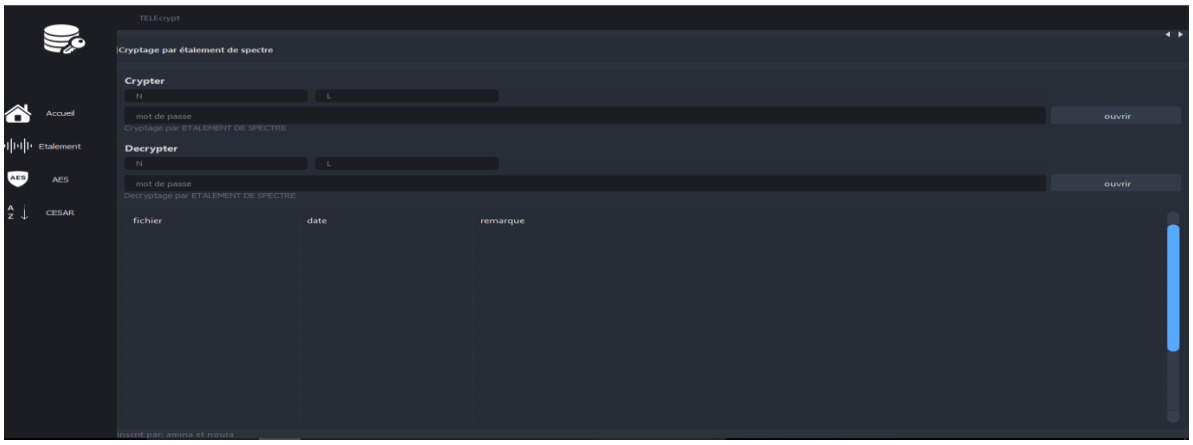


-Ensuite, on ajouter un stacked widget pour créer plusieurs pages sur notre fenêtre.

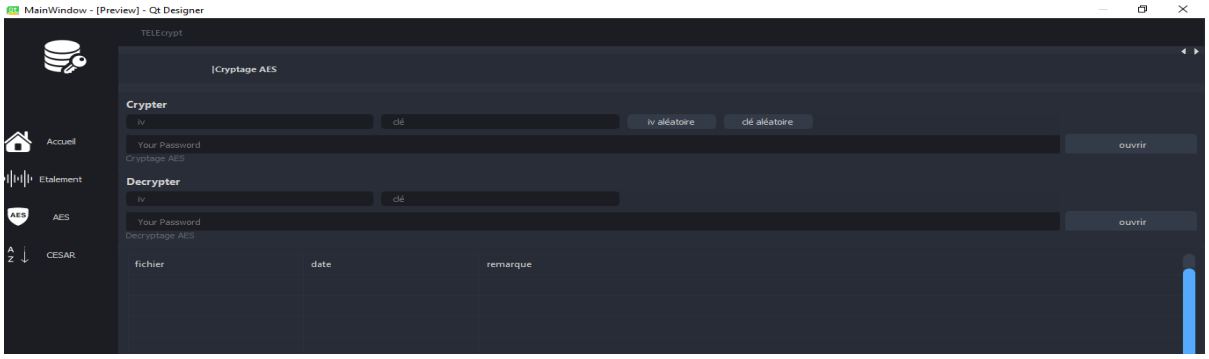


-Les trois frames correspondent au cryptage, décryptage et tableau d'affichage, respectivement. Les trois pages relatives à la sécurisation des données s'affichent comme suit :

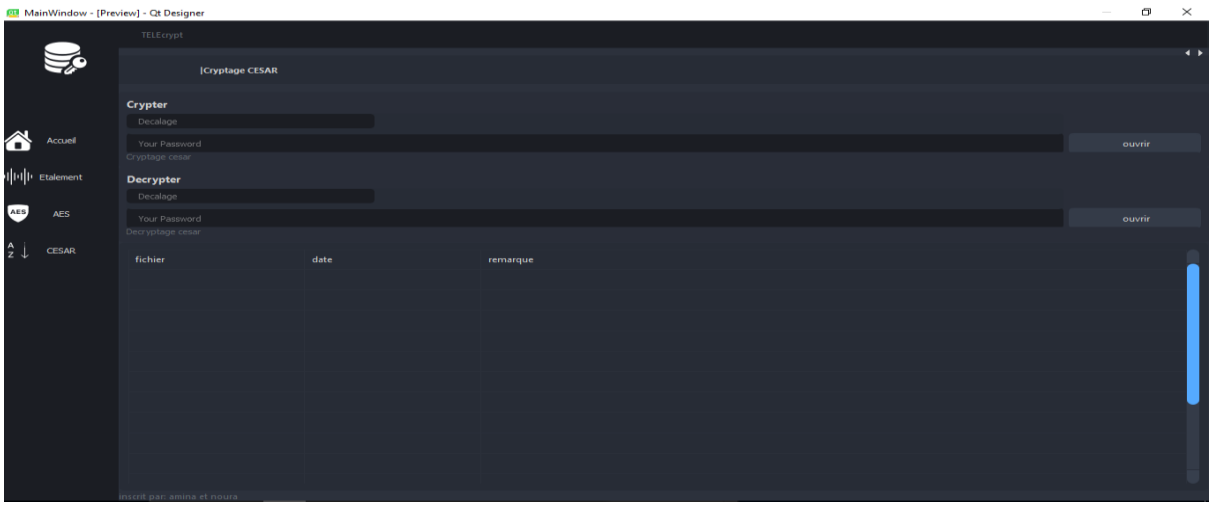
1 ere Page (étalement de spectre):



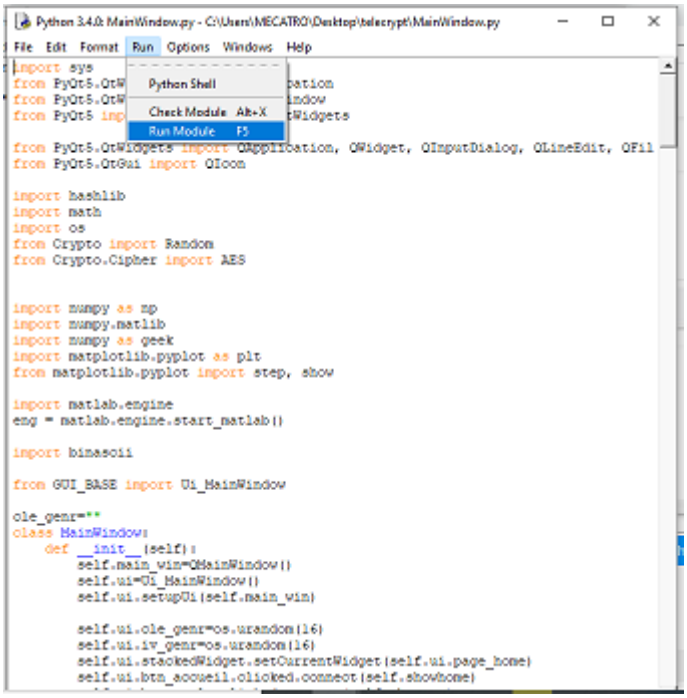
2 ème Page (AES):



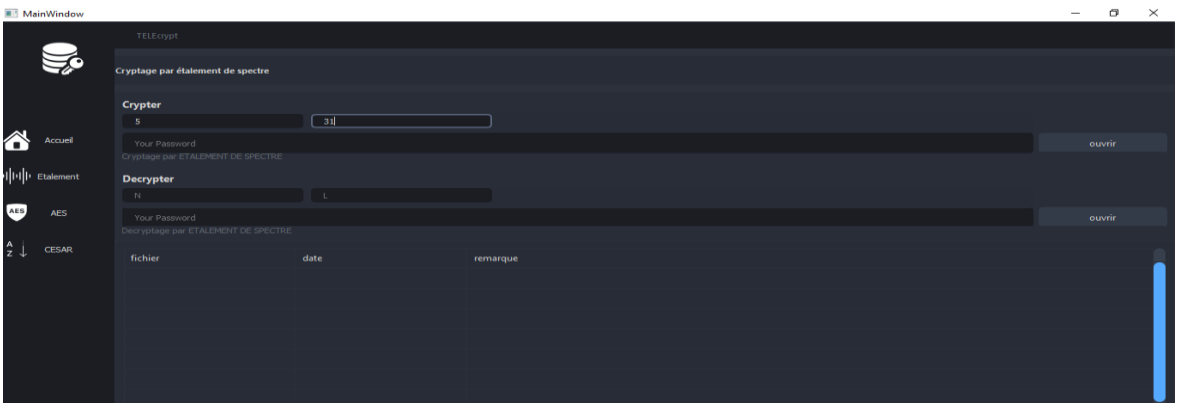
3 ème Page (César):

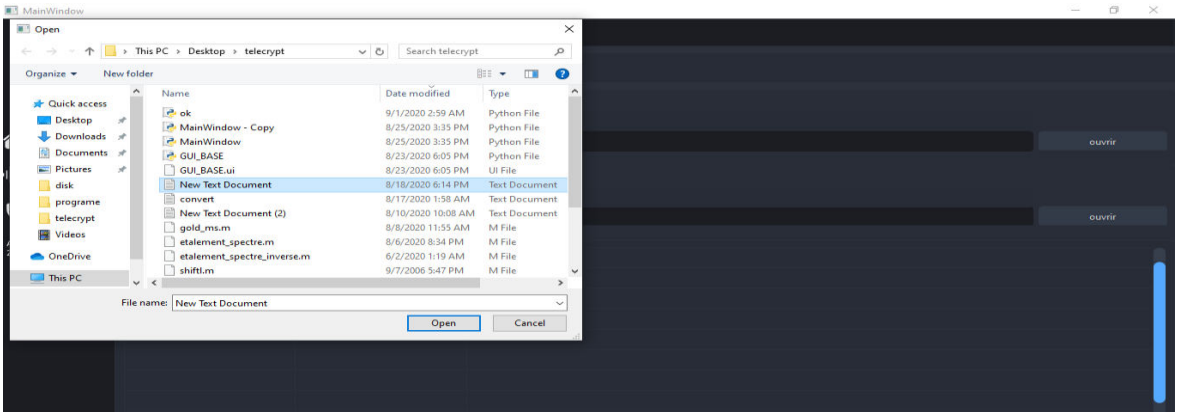


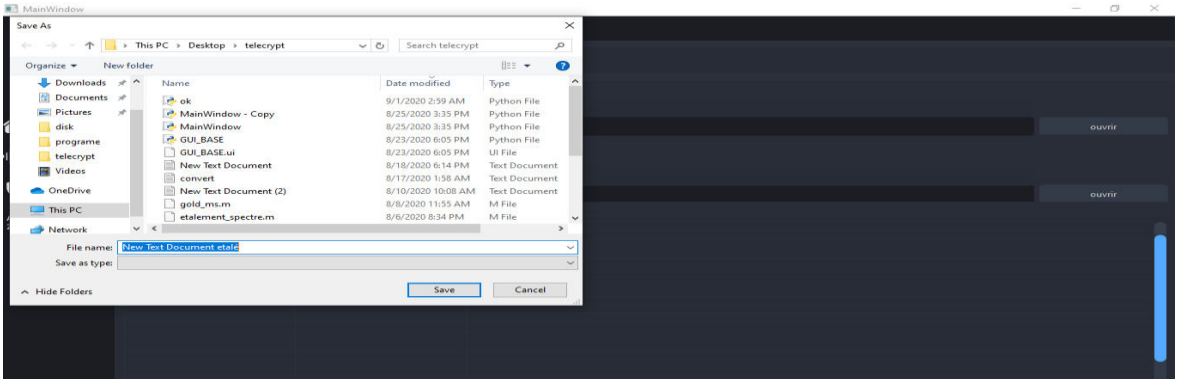
Le code source du logiciel telecrypt et sa page d'accueil sont présentés ci-après :

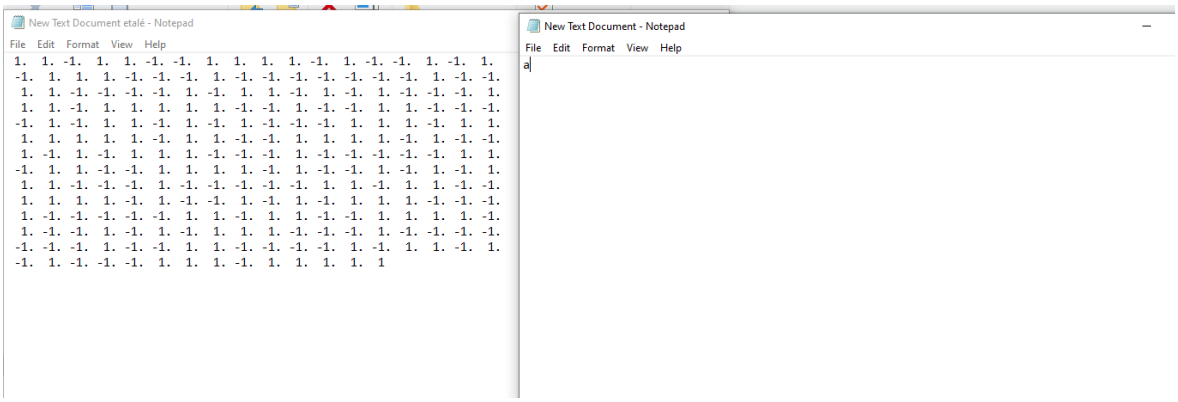


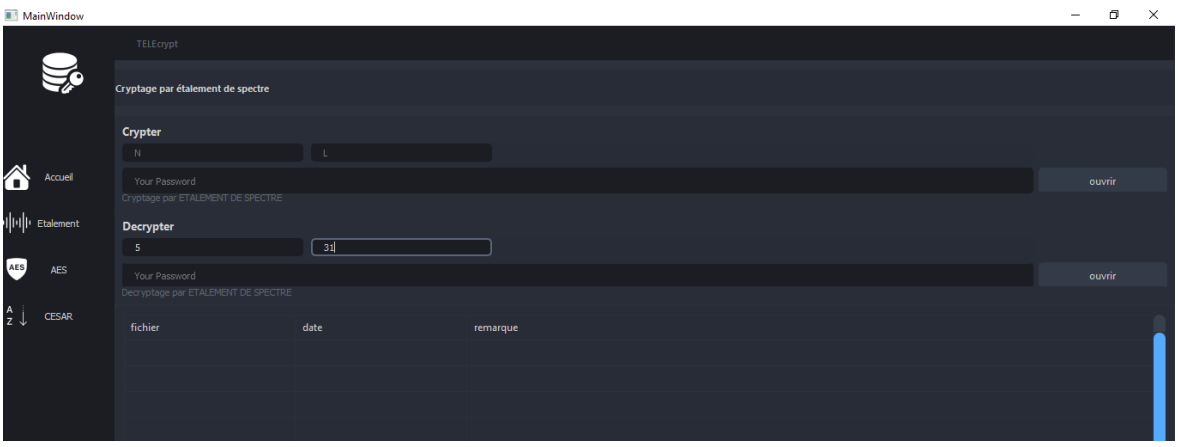
Un exemple de sécurisation de l'information transmise par la technique de l'étalement de spectre et décrit par les figure suivantes :

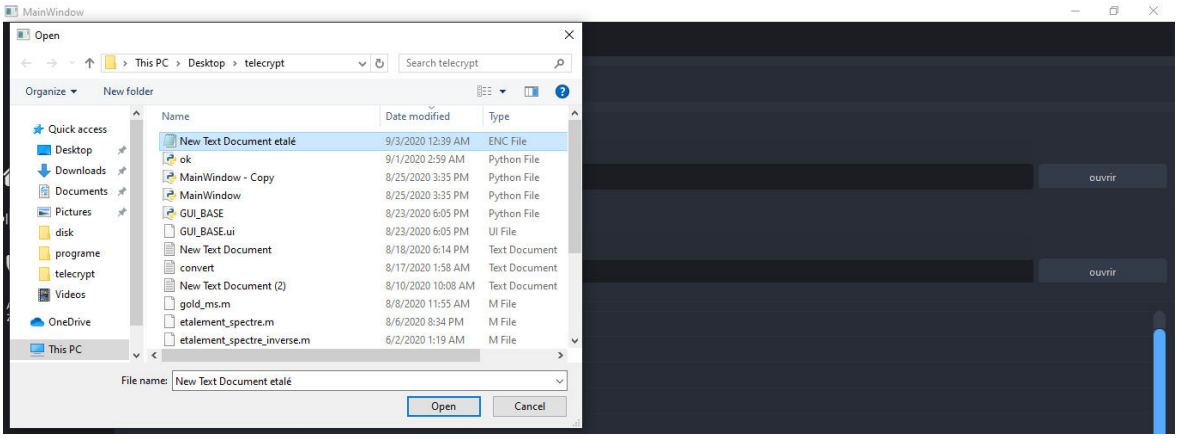


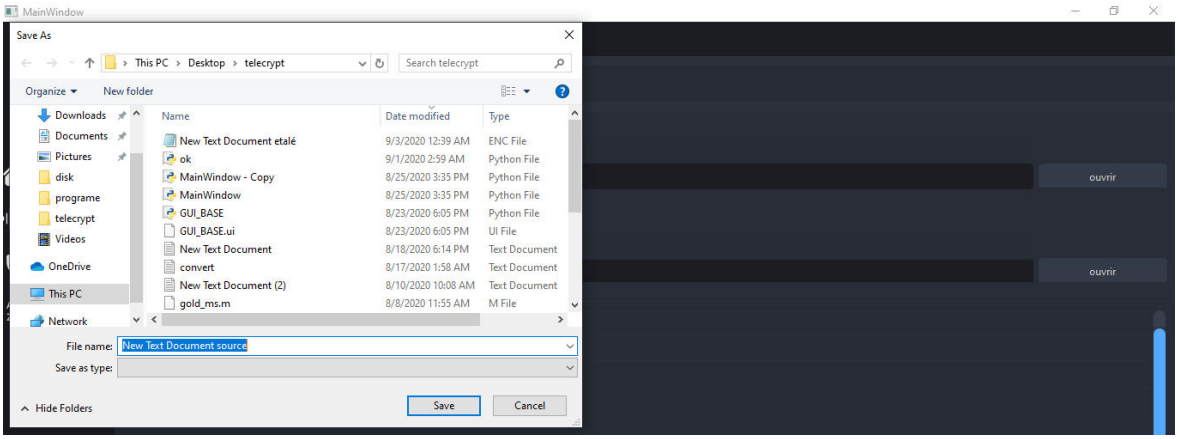

MainWindow
Open
This PC › Desktop › telecrypt
Search telecrypt
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
disk
programe
telecrypt
Videos
OneDrive
This PC
Name
Date modified
Type
ok 9/1/2020 2:59 AM Python File
MainWindow - Copy 8/25/2020 3:35 PM Python File
MainWindow 8/25/2020 3:35 PM Python File
GUI_BASE 8/23/2020 6:05 PM Python File
GUI_BASE.ui 8/23/2020 6:05 PM UI File
New Text Document 8/18/2020 6:14 PM Text Document
convert 8/17/2020 1:58 AM Text Document
New Text Document (2) 8/10/2020 10:08 AM Text Document
gold_ms.m 8/8/2020 11:55 AM M File
etalement_spectre.m 8/6/2020 8:34 PM M File
etalement_spectre_inverse.m 6/2/2020 1:19 AM M File
shiftl.m 9/7/2006 5:47 PM M File
File name: New Text Document
Open
Cancel
ouvrir
ouvrir

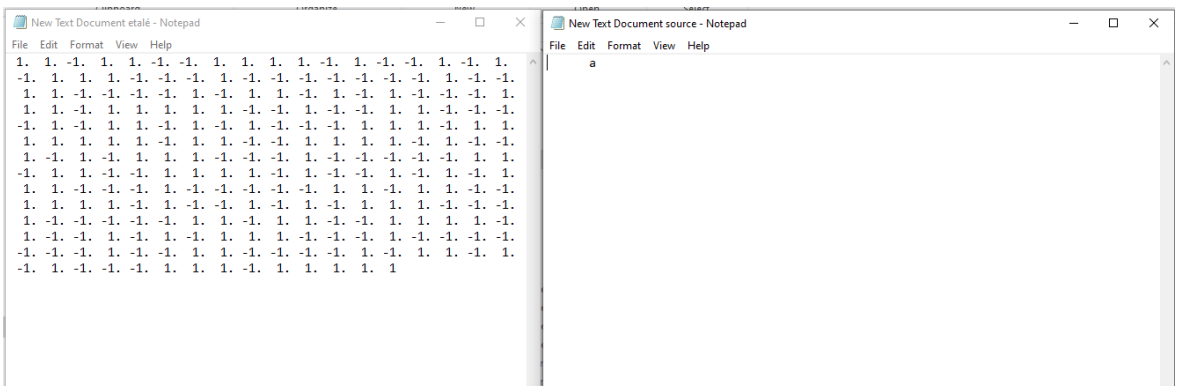

MainWindow
Save As
This PC › Desktop › telecrypt
Search telecrypt
Organize
New folder
Downloads
Documents
Pictures
disk
programe
telecrypt
Videos
OneDrive
This PC
Network
Name
Date modified
Type
ok 9/1/2020 2:59 AM Python File
MainWindow - Copy 8/25/2020 3:35 PM Python File
MainWindow 8/25/2020 3:35 PM Python File
GUI_BASE 8/23/2020 6:05 PM Python File
GUI_BASE.ui 8/23/2020 6:05 PM UI File
New Text Document 8/18/2020 6:14 PM Text Document
convert 8/17/2020 1:58 AM Text Document
New Text Document (2) 8/10/2020 10:08 AM Text Document
gold_ms.m 8/8/2020 11:55 AM M File
etalement_spectre.m 8/6/2020 8:34 PM M File
File name: New Text Document etalé
Save as type:
Hide Folders
Save
Cancel
ouvrir
ouvrir

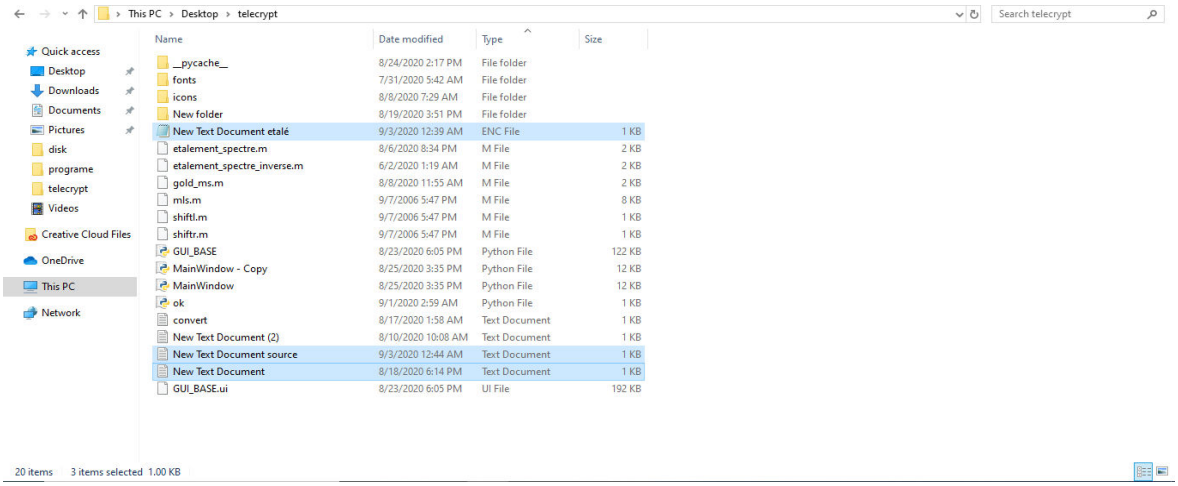

New Text Document etalé - Notepad
File Edit Format View Help
1. 1. -1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1.
-1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. -1. 1. -1. -1.
1. 1. -1. -1. -1. -1. 1. -1. 1. 1. -1. 1. -1. 1. -1. -1. -1. 1.
1. 1. -1. 1. 1. 1. 1. -1. -1. 1. -1. -1. 1. 1. -1. -1. -1.
-1. 1. -1. 1. 1. -1. 1. -1. 1. -1. -1. -1. 1. 1. 1. -1. 1. 1.
1. 1. 1. 1. 1. -1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1.
1. -1. 1. -1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. 1. 1.
-1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1. -1. 1.
1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. 1. 1. -1. 1. 1. -1. -1.
1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1. -1. 1. 1. 1. -1. -1. -1.
1. -1. -1. -1. -1. -1. 1. 1. -1. 1. 1. -1. -1. 1. 1. 1. 1. -1.
1. -1. -1. 1. -1. 1. -1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1.
-1. -1. -1. 1. -1. -1. 1. 1. -1. -1. -1. -1. 1. -1. 1. 1. -1. 1.
-1. 1. -1. -1. -1. 1. 1. 1. -1. 1. 1. 1. 1. 1
New Text Document - Notepad
File Edit Format View Help
a


MainWindow
TELEcrypt
Cryptage par étalement de spectre
Crypter
N
L
Your Password
ouvrir
Cryptage par ETALEMENT DE SPECTRE
Decrypter
5
31
Your Password
ouvrir
Decryptage par ETALEMENT DE SPECTRE
fichier
date
remarque
Accueil
Etalement
AES
CESAR


MainWindow
Open
This PC › Desktop › telecrypt
Search telecrypt
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
disk
programe
telecrypt
Videos
OneDrive
This PC
Name
Date modified
Type
New Text Document etalé 9/3/2020 12:39 AM ENC File
ok 9/1/2020 2:59 AM Python File
MainWindow - Copy 8/25/2020 3:35 PM Python File
MainWindow 8/25/2020 3:35 PM Python File
GUI_BASE 8/23/2020 6:05 PM Python File
GUI_BASE.ui 8/23/2020 6:05 PM UI File
New Text Document 8/18/2020 6:14 PM Text Document
convert 8/17/2020 1:58 AM Text Document
New Text Document (2) 8/10/2020 10:08 AM Text Document
gold_ms.m 8/8/2020 11:55 AM M File
etalement_spectre.m 8/6/2020 8:34 PM M File
etalement_spectre_inverse.m 6/2/2020 1:19 AM M File
File name: New Text Document etalé
Open
Cancel
ouvrir
ouvrir


MainWindow
Save As
This PC › Desktop › telecrypt
Search telecrypt
Organize
New folder
Downloads
Documents
Pictures
disk
programe
telecrypt
Videos
OneDrive
This PC
Network
Name
Date modified
Type
New Text Document etalé 9/3/2020 12:39 AM ENC File
ok 9/1/2020 2:59 AM Python File
MainWindow - Copy 8/25/2020 3:35 PM Python File
MainWindow 8/25/2020 3:35 PM Python File
GUI_BASE 8/23/2020 6:05 PM Python File
GUI_BASE.ui 8/23/2020 6:05 PM UI File
New Text Document 8/18/2020 6:14 PM Text Document
convert 8/17/2020 1:58 AM Text Document
New Text Document (2) 8/10/2020 10:08 AM Text Document
gold_ms.m 8/8/2020 11:55 AM M File
File name: New Text Document source
Save as type:
Hide Folders
Save
Cancel
ouvrir
ouvrir


New Text Document etalé - Notepad
File Edit Format View Help
1. 1. -1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1.
-1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. -1. -1. 1. -1. -1.
1. 1. -1. -1. -1. -1. 1. -1. 1. 1. -1. 1. -1. 1. -1. -1. -1. 1.
1. 1. -1. 1. 1. 1. 1. 1. -1. -1. 1. -1. -1. 1. 1. -1. -1. -1.
-1. 1. -1. 1. 1. -1. 1. -1. 1. -1. -1. -1. 1. 1. 1. -1. 1. 1.
1. 1. 1. 1. 1. -1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1.
1. -1. 1. -1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. 1. 1.
-1. 1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1. -1. 1.
1. 1. -1. -1. -1. 1. -1. -1. -1. -1. -1. 1. 1. -1. 1. 1. -1. -1.
1. 1. 1. 1. -1. 1. -1. -1. 1. -1. 1. -1. 1. 1. 1. -1. -1. -1.
1. -1. -1. -1. -1. -1. 1. 1. 1. -1. 1. 1. -1. -1. 1. 1. 1. -1.
1. -1. -1. 1. -1. 1. -1. 1. 1. 1. -1. -1. -1. 1. -1. -1. -1. -1.
-1. -1. -1. 1. -1. -1. 1. 1. 1. -1. -1. -1. -1. 1. -1. 1. 1. -1. 1.
-1. 1. -1. -1. -1. 1. 1. 1. -1. 1. 1. 1. 1. 1
New Text Document source - Notepad
File Edit Format View Help
a



IV.5 Conclusion

Afin de faciliter la tâche à l'utilisateur, nous avons pensé à organiser notre projet de fin d'études sous forme d'un logiciel, appelé telecrypt, de telle sorte de permettre une manipulation souple des paramètres de transmission/réception de données.

De point de vue émetteur, le NAS est créé au niveau de Rasbperry Pi 4. L'utilisateur écrit son message dans un fichier texte puis il l'enregistre. A partir des boutons réalisés, il choisira la technique de sécurisation voulue. Le logiciel demande à l'utilisateur d'enregistrer le message étalé et/ou crypté, soit par exemple message_secure, qui est automatiquement sauvegardé dans le NAS.

De point de vue récepteur, l'utilisateur se connecte sur le réseau WiFi appelé Pi_AP. A travers le logiciel telecrypt, il ouvre fichier sécurisé message_secure et il reconstituera le message de source. Bien évidemment, ceci n'est plus possible que lorsque le récepteur connait parfaitement la méthode de sécurisation adoptée en émission et ses paramètres que ce soit l'étalement de spectre, le chiffrement César, le codage AES ou les trois modes à la fois. On se trouve ainsi avec un système de communication numérique full duplex, sécurisé, à utilisateurs multiples à base de Raspberry.

Conclusion générale

Conclusion générale

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant notre cycle long de formation universitaire, en particulier le master en systèmes des télécommunications.

Dans ce contexte, le but de notre projet était de réaliser un système sécurisé de transmission de données au tour de Raspberry Pi manipulé à travers un logiciel conçu pour ce but : Telecrypt. Les émetteurs et récepteurs ayant accès à notre réseau WiFi, appelé Pi_AP, ont la possibilité d'émettre et recevoir des fichiers textes à travers le serveur NAS. La sécurisation doit être établie au début de chaque transfert de données qui sont supposées, dans cette première version du projet, d'être textuelles. Le protocole de sécurisation inclut un des trois modes implémentés pour cet effet ou les trois à la fois, à savoir : l'étalement de spectre à séquence directe par le code Gold, le chiffrement César et le cryptage AES.

Lors de l'achèvement du projet, nous avons rencontré plusieurs obstacles et problèmes qui ont retardé son achèvement. L'un des obstacles les plus sérieux est le manque d'équipement dans les magasins ou leur pénurie. Il fallait parfois attendre un certain temps pour que la commande du matériel arrive.

Nous avons également rencontré le problème très gênant de faible débit d'Internet. Cette faiblesse de réseau a beaucoup tardé notre projet suite à la difficulté de chargement des logiciels et des commandes appropriés au système réalisé, en plus de l'impossibilité d'organiser des réunions distants avec l'encadreur en utilisant les outils de visioconférence comme Zoom.

Un autre problème qui est global et concerne le monde entier est l'obligation de rester confinés à cause de la pandémie Covid-19. En plus de la pression due à la fermeture et des effets psychologiques qu'ils ont suit ; il n'était plus possible de déplacer, de se réunir et de travailler en équipe. Il était aussi impossible de recevoir le support scientifique, documentaire mais aussi moral qui est indispensable afin de munir un projet de fin de cycle aussi important qu'un master académique en télécommunications.

Bien que notre travail a bien abouti à ses objectifs déclarés dans le cahier de charge; il y a quelques points qui nécessitent des améliorations comme présenté dans les perspectives suivantes :

Perspectives de développement du projet:

- Plusieurs organisations et entreprises peuvent exploiter le projet développé de différentes manières. Les perspectives de développement de la plateforme sont également nombreuses, notamment par rapport aux besoins des universités et écoles supérieures.
- Nous travaillerons à la réalisation de ce projet pour le vendre, le projet terminé peut être utilisé pour élever plusieurs niveaux.
- Nous pouvons y ajouter en créant un site Web divisé en trois parties

  - La première partie est réservée à l'administration, elle contient toutes les informations administratives des documents et des fichiers pour relier toute l'administration entre elles.

  - La seconde partie permet aux professeurs d'échanger des informations privées et leurs activités de manière distante et confidentielle.

  - La troisième partie est destinée aux enseignants et aux étudiants afin de faciliter le téléchargement de toutes les leçons sans frais pour l'étudiant.
- La dernière suggestion est d'augmenter la portée et d'améliorer la sécurité en adoptant des techniques d’étalement de spectre et de cryptographie beaucoup plus robustes.

Bibliographié

Bibliographies

[1] https://xcotton.pagesperso-orange.fr/electron/generalites_sur_la_transmission_des_donnees.pdf

[2] https://pmb.univ-saida.dz/butecopac/doc_num.php?explnum_id=56776

[3] http://cte.univ-setif.dz/coursenligne/lahcene/chap2.htmll

[4] http://dspace.univ-tlemcen.dz/bitstream/112/6836/1/Etude-comparative-entre-la-cryptographie.pdf

[5] https://fr.wikipedia.org/wiki/Chiffrement

[6] https://www.futura-sciences.com/sciences/questions-reponses/mathematiques-quest-ce-chiffrement-cesar

[7] https://www.google.com/search?q=chiffrement+par+d%C3%A9calage

[8] https://www.securiteinfo.com/cryptographie/aes.shtml

[10] https://pmb.univ-saida.dz/butecopac/doc_num.php?explnum_id=56776

[11] G.Li,"Physical Layer Design for a Spread Spectrum Wireless LAN," M. Sc. Thesis in electrical engineering, Faculty of the Virginia Polytechnic Institute and State University,Blacksburg, Virginia, Sep.1996.

[12] Homier E. A. and Scholtz, R. A. "Rapid acquisition of ultra-wideband signals in the dense multipath channel," in IEEE Conference on Ultra Widband Systems and Technology (UWBST),pp.105-109,May 2002

[13]http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm

[14] Spread Spectrum Techniques, Delft University of Technology Circuits & Systems Group, 2000. Available at http://cobalt.et.tudelft.n1/wissce/techn/techniques.html.

[15]https://dl.ummto.dz/bitstream/handle/ummto/6734/BouaklineNassim_BenchabaAghiles.pdf?sequence=1.

[16] https://www.framboise314.fr/raspberry-pi-4-4-nouveautes-qui-vont-vous-faire-craquer/

[17] https://fr.wikibooks.org/wiki/Programmation_Qt/Qt_Designer

[18] http://perso.unifr.ch/ales.janka/analnum/intro_matlab_fr.pdf

[19] https://www.youtube.com/watch?v=2SSQNcktE-8

[20] https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up?fbclid=IwAR1_GE5hG_L-sB43bI6AHjf383xHMwmX9P0cNULiA8gvv9-k1rZnVbU-1W8

[21] https://doc.ubuntu-fr.org/hostapd

[22] http://www.fsg.rnu.tn/imgsite/cours/samba.pdf

[23] https://fr.wikibooks.org/wiki/Programmation_Python/Introduction

[24]https://fr.wikibooks.org/wiki/Programmation_Qt/Qt_Designer#:~:text=Qt%20designer%20est%20une%20s%C3%A9rie,utilisables%20comme%20des%20classes%20normales.

[25] http://perso-laris.univ-angers.fr/~delanoue/istia/calcul_numerique/td1.pdf

Annexes

Annexes A : Code source Matlab du code Gold

```
function [gold_code] = gold_ms(n,L)
MLS=zeros(L,2^n-1);
gold=zeros(L,2^n-1);
if n==5
    u1=mls(31,1);
    u2=mls(31,4);
end;
if n==7
    u1=mls(127,3);
    u2=mls(127,5);
end;
if n==8
    u1=mls(255,2);
    u2=mls(255,15);
end;
if n==9
     u1=mls(511,1);
     u2=mls(551,3);
end;
    for i=1:L
        h1(:,1)=(u1(:,1)+1)/2;
        h2(:,1)=(u2(:,1)+1)/2;
%
gold(:,i)=xor(h1(:,1),shiftl(h2(:,1),i-1));
end;
gold_code=gold*2-1;
```

code shift:

```
function [dvec]=shiftr(C,shift);
s=length(C);
% for shift=1:(L-1)
% if ~isempty(shift),
% shift1=shift;
%    shift1=rem(shift1, length(C));
j=1;
 for i=1:(shift)
  dvec(i)=C(s-shift+i);
 % j=j+1;
end;
for j=(shift+1):s
```

```
  dvec(j)=C(j-shift);
end;
dvec=dvec';
```

code mls :

```
function [mlsc]=mls(L,whichseq);
 baseVal=2;
 mls=zeros(L,L);
   n=log2(L+1);
 shift=L;
 register=ones(n,1);
if baseVal==2,
switch n,
case 2, tap(1).No=[1,2];
case 3, tap(1).No=[1,3];
          tap(2).No=[2,3];
case 4, tap(1).No=[1,4];
          tap(2).No=[3,4];
case 5, tap(1).No=[2,5];
          tap(2).No=[3,5];
          tap(3).No=[1,2,3,5];
          tap(4).No=[2,3,4,5];
          tap(5).No=[1,2,4,5];
          tap(6).No=[1,3,4,5];
case 6, tap(1).No=[1,6];
          tap(2).No=[5,6];
          tap(3).No=[1,2,5,6];
          tap(4).No=[1,4,5,6];
          tap(5).No=[1,3,4,6];
          tap(6).No=[2,3,5,6];
case 7, tap(1).No=[1,7];
          tap(2).No=[6,7];
          tap(3).No=[3,7];
          tap(4).No=[4,7];
          tap(5).No=[1,2,3,7];
          tap(6).No=[4,5,6,7];
          tap(7).No=[1,2,5,7];
          tap(8).No=[2,5,6,7];
          tap(9).No=[2,3,4,7];
          tap(10).No=[3,4,5,7];
          tap(11).No=[1,3,5,7];
          tap(12).No=[2,4,6,7];
```

```
          tap(13).No=[1,3,6,7];
          tap(14).No=[1,4,6,7];
          tap(15).No=[2,3,4,5,6,7];
          tap(16).No=[1,2,3,4,5,7];
          tap(17).No=[1,2,4,5,6,7];
          tap(18).No=[1,2,3,5,6,7];
case 8,  tap(1).No=[1,2,7,8];
          tap(2).No=[1,6,7,8];
          tap(3).No=[1,3,5,8];
          tap(4).No=[3,5,7,8];
          tap(5).No=[2,3,4,8];
          tap(6).No=[4,5,6,8];
          tap(7).No=[2,3,5,8];
          tap(8).No=[3,5,6,8];
          tap(9).No=[2,3,6,8];
          tap(10).No=[2,5,6,8];
          tap(11).No=[2,3,7,8];
          tap(12).No=[1,5,6,8];
          tap(13).No=[1,2,3,4,6,8];
          tap(14).No=[1,2,3,6,7,8];
          tap(15).No=[1,2,5,6,7,8];
case 9,  tap(1).No=[4,9];
          tap(2).No=[5,9];
          tap(3).No=[3,4,6,9];
          tap(4).No=[3,5,6,9];
          tap(5).No=[4,5,8,9];
          tap(6).No=[1,4,5,9];
          tap(7).No=[1,4,8,9];
          tap(8).No=[1,5,8,9];
          tap(9).No=[2,3,5,9];
          tap(10).No=[4,6,7,9];
          tap(11).No=[5,6,8,9];
          tap(12).No=[1,3,4,9];
          tap(13).No=[2,7,8,9];
          tap(14).No=[1,2,7,9];
          tap(15).No=[2,4,7,9];
          tap(16).No=[2,5,7,9];
          tap(17).No=[2,4,8,9];
          tap(18).No=[1,5,7,9];
          tap(19).No=[1,2,4,5,6,9];
          tap(20).No=[3,4,5,7,8,9];
```

```
            tap(21).No=[1,3,4,6,7,9];
            tap(22).No=[2,3,5,6,8,9];
            tap(23).No=[3,5,6,7,8,9];
            tap(24).No=[1,2,3,4,6,9];
            tap(25).No=[1,5,6,7,8,9];
            tap(26).No=[1,2,3,4,8,9];
            tap(27).No=[1,2,3,7,8,9];
            tap(28).No=[1,2,6,7,8,9];
            tap(29).No=[1,3,5,6,8,9];
            tap(30).No=[1,3,4,6,8,9];
            tap(31).No=[1,2,3,5,6,9];
            tap(32).No=[3,4,6,7,8,9];
            tap(33).No=[2,3,6,7,8,9];
            tap(34).No=[1,2,3,6,7,9];
            tap(35).No=[1,4,5,6,8,9];
            tap(36).No=[1,3,4,5,8,9];
            tap(37).No=[1,3,6,7,8,9];
            tap(38).No=[1,2,3,6,8,9];
            tap(39).No=[2,3,4,5,6,9];
            tap(40).No=[3,4,5,6,7,9];
            tap(41).No=[2,4,6,7,8,9];
            tap(42).No=[1,2,3,5,7,9];
            tap(43).No=[2,3,4,5,7,9];
            tap(44).No=[2,4,5,6,7,9];
            tap(45).No=[1,2,4,5,7,9];
            tap(46).No=[2,4,5,6,7,9];
            tap(47).No=[1,3,4,5,6,7,8,9];
            tap(48).No=[1,2,3,4,5,6,8,9];
case 10, tap(1).No=[3,10];
             tap(2).No=[7,10];
             tap(3).No=[2,3,8,10];
             tap(4).No=[2,7,8,10];
             tap(5).No=[1,3,4,10];
             tap(6).No=[6,7,9,10];
             tap(7).No=[1,5,8,10];
             tap(8).No=[2,5,9,10];
             tap(9).No=[4,5,8,10];
             tap(10).No=[2,5,6,10];
             tap(11).No=[1,4,9,10];
             tap(12).No=[1,6,9,10];
             tap(13).No=[3,4,8,10];
```

```
tap(14).No=[2,6,7,10];
tap(15).No=[2,3,5,10];
tap(16).No=[5,7,8,10];
tap(17).No=[1,2,5,10];
tap(18).No=[5,8,9,10];
tap(19).No=[2,4,9,10];
tap(20).No=[1,6,8,10];
tap(21).No=[3,7,9,10];
tap(22).No=[1,3,7,10];
tap(23).No=[1,2,3,5,6,10];
tap(24).No=[4,5,7,8,9,10];
tap(25).No=[2,3,6,8,9,10];
tap(26).No=[1,2,4,7,8,10];
tap(27).No=[1,5,6,8,9,10];
tap(28).No=[1,2,4,5,9,10];
tap(29).No=[2,5,6,7,8,10];
tap(30).No=[2,3,4,5,8,10];
tap(31).No=[2,4,6,8,9,10];
tap(32).No=[1,2,4,6,8,10];
tap(33).No=[1,2,3,7,8,10];
tap(34).No=[2,3,7,8,9,10];
tap(35).No=[3,4,5,8,9,10];
tap(36).No=[1,2,5,6,7,10];
tap(37).No=[1,4,6,7,9,10];
tap(38).No=[1,3,4,6,9,10];
tap(39).No=[1,2,6,8,9,10];
tap(40).No=[1,2,4,8,9,10];
tap(41).No=[1,4,7,8,9,10];
tap(42).No=[1,2,3,6,9,10];
tap(43).No=[1,2,6,7,8,10];
tap(44).No=[2,3,4,8,9,10];
tap(45).No=[1,2,4,6,7,10];
tap(46).No=[3,4,6,8,9,10];
tap(47).No=[2,4,5,7,9,10];
tap(48).No=[1,3,5,6,8,10];
tap(49).No=[3,4,5,6,9,10];
tap(50).No=[1,4,5,6,7,10];
tap(51).No=[1,3,4,5,6,7,8,10];
tap(52).No=[2,3,4,5,6,7,9,10];
tap(53).No=[3,4,5,6,7,8,9,10];
tap(54).No=[1,2,3,4,5,6,7,10];
```

```
            tap(55).No=[1,2,3,4,5,6,9,10];
            tap(56).No=[1,4,5,6,7,8,9,10];
            tap(57).No=[2,3,4,5,6,8,9,10];
            tap(58).No=[1,2,4,5,6,7,8,10];
            tap(59).No=[1,2,3,4,6,7,9,10];
            tap(60).No=[1,3,4,6,7,8,9,10];
case 11, tap(1).No=[9,11];
              tap(2).No=[2,5,8,11];
              tap(3).No=[2,3,7,11];
              tap(4).No=[2,3,10,11];
              tap(5).No=[1,5,6,11];
              tap(6).No=[1,3,5,11];
              tap(7).No=[1,4,9,11];
              tap(8).No=[2,6,8,11];
              tap(9).No=[3,8,9,11];
case 12, tap(1).No=[6,8,11,12];
              tap(2).No=[2,5,8,11];
              tap(3).No=[2,3,7,11];
              tap(4).No=[2,3,10,11];
              tap(5).No=[1,5,6,11];
              tap(6).No=[1,3,5,11];
              tap(7).No=[1,4,9,11];
              tap(8).No=[2,6,8,11];
              tap(9).No=[3,8,9,11];
case 13, tap(1).No=[9,10,12,13];
case 14, tap(1).No=[4,8,13,14];
case 15, tap(1).No=[14,15];
case 16, tap(1).No=[4,13,15,16];
case 17, tap(1).No=[14,17];
case 18, tap(1).No=[11,18];
case 19, tap(1).No=[14,17,18,19];
case 20, tap(1).No=[17,20];
case 21, tap(1).No=[19,21];
case 22, tap(1).No=[21,22];
case 23, tap(1).No=[18,23];
case 24, tap(1).No=[17,22,23,24];
case 25, tap(1).No=[22,25];
case 26, tap(1).No=[20,24,25,26];
case 27, tap(1).No=[22,25,26,27];
case 28, tap(1).No=[25,28];
case 29, tap(1).No=[27,29];
```

```
case 30, tap(1).No=[7,28,29,30];
otherwise error(sprintf('M-sequence %.0f^%.0f is not defined',baseVal,n))
end;
ms=zeros(L,1);
weights=zeros(1,n);
if baseVal==2,
    weights(tap(whichseq).No)=1;
end;
for i=1:L
  ms(i)=rem(weights*register+baseVal,baseVal);
  register=[ms(i);register(1:n-1)];
end
 ms=ms(:);
mls(:,1)=ms(:);
for shift=1:(L-1)
 shift1=shift;
 shift1=rem(shift1, length(ms));
 mls(:,shift+1)=[ms(shift1+1:end); ms(1:shift1)];
end;
mls(:,1:L);
mls=mls*2-1;
mlsc=mls;
end;
```