MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA 1

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER en informatique

THEME :

Comparaison des difficultés de craquage d'attaque du type RSA et TKIP

Spécialité : Sécurité des systèmes d'informations

Réalisé par : Traoré Ramata

Promoteur : Mr Benyahia Mohamed

Jury : Mr Ould Khaou

Mm Bey Fella

Année universitaire : 2022-2023

Remerciement

Je tiens à remercier en première point mon créateur de m’avoir permis de respirer jusqu’au jour j, à exprimer ma profonde gratitude et mes sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce mémoire de fin d'études. Leur soutien, leur expertise et leurs encouragements ont été essentiels tout au long de ce parcours académique.

Un grand remerciement a mon promoteur de mémoire Mr Benyahia, pour son orientation, ses précieux conseils et son engagement constant. Sa disponibilité et son expertise ont grandement enrichi ce travail de recherche.

Je tiens à adresser mes remerciements aux membres du jury Mm Bey Fella et Mr Ould Khaou, pour leur temps, leur attention et leurs précieuses remarques lors de la soutenance de ce mémoire.

Un grand merci à la communauté de blida, qui m’ont accordé leur soutien tout au long de cette aventure académique. Leurs discussions fructueuses et leur présence chaleureuse ont rendu cette expérience d'apprentissage encore plus enrichissante.

Je n'oublierai jamais le soutien inconditionnel de ma famille, en particulier de mes parents (Aminata, Idrissa et Lamine) et de ma sœur de cœur (Mariam NYSK) qui ont toujours cru en moi et m'ont encouragé à persévérer dans mes études. Leur amour et leur soutien ont été la force motrice qui m'a permis de surmonter les obstacles et d'atteindre cet objectif.

Merci à tous (surtout la cité 5) pour votre bienveillance, votre collaboration et votre appui tout au long de cette belle aventure universitaire. Votre soutien a été précieux, et je suis fier de partager ce mémoire avec vous.

SOMMAIRE

LISTES DES FIGURES

LISTE DES TABLEAUX

Résumé

Ce projet vise à tester, analyser et comparer la résistance de la méthode RSA en l'état actuel du progrès mathématique et technologique et du protocole de chiffrement TKIP. Bien que le RSA soit un algorithme de cryptographie asymétrique qui utilise deux clés (une clé publique pour le chiffrement et une clé privée pour le déchiffrement), et TKIP un algorithme de cryptographie symétrique qui utilise une même clé pour le chiffrement et le déchiffrement, ils sont vulnérables à de nombreuses attaques qui cherchent à les compromettre.

Abstract

This project aims to test, analyze and compare the resistance of the RSA method in the current state of mathematical and technological progress and of the TKIP encryption protocol. Although RSA is an asymmetric cryptography algorithm that uses two keys (a public key for encryption and a private key for decryption), and TKIP is a symmetric cryptography algorithm that uses the same key for encryption and decryption, they are vulnerable to many attacks that seek to compromise them.

ملخص

يهدف هذا المشروع إلى اختبار وتحليل ومقارنة مقاومة طريقة RSA في الحالة الحالية للتقدم الرياضي والتكنولوجي وبروتوكول تشفير TKIP. على الرغم من أن RSA (مفتاح عبارة عن خوارزمية تشفير غير متماثلة تستخدم مفتاحين TKIP عام للتشفير ومفتاح خاص لفك التشفير) ، و عبارة عن خوارزمية تشفير متماثل تستخدم نفس المفتاح للتشفير وفك التشفير ، إلا أنها عرضة للعديد من الهجمات التي تسعى إلى للتنازل عنها.

Introduction Générale

La rapide évolution des réseaux informatiques, qu'ils soient privés ou publics, engendre un volume croissant de données sauvegardées et transmises. Cette croissance exponentielle a créé de nouveaux besoins en matière de sécurité, d'autant plus que les entreprises dépendent de plus en plus de leur système informatique. La sécurité est ainsi devenue une préoccupation majeure.

Dans ce contexte, la cryptographie et le chiffrement sont indispensables pour assurer une sécurité informatique de pointe. La cryptographie offre des mécanismes pour assurer la confidentialité, l'authenticité et l'intégrité des données. Elle se divise en deux grandes familles : la cryptographie symétrique et la cryptographie asymétrique. La cryptographie moderne a été marquée par la proposition d'une méthode de chiffrement des données basée sur un schéma de Feistel : le Data Encryptions Standard (DEPUIS). Cet algorithme a été remplacé quelques années plus tard par un nouveau standard plus robuste basé sur une structure algébrique, plus précisément sur un réseau de substitution- permutation : l'Advanced Encryption Standard (AES).

Le protocole de chiffrement TKIP (Temporal Key Integrity Protocol) a été développé pour renforcer la sécurité des réseaux sans fil basés sur la norme IEEE 802.11. Il a été utilisé comme une mesure intermédiaire entre WEP (Wired Equivalent Privacy) et WPA (Wi-Fi Protected Access). TKIP implémente des techniques telles que la génération dynamique de clés, la rotation régulière des clés et la vérification de l'intégrité des données chiffrées pour améliorer la sécurité des clés utilisées dans le chiffrement des données sans fil.[14]

Ces deux méthodes de chiffrement, AES et TKIP, font partie des systèmes cryptographiques symétriques, ce qui signifie que deux parties souhaitant communiquer doivent partager le même secret, appelé clé secrète, pour chiffrer et déchiffrer les données. Bien que ces systèmes soient performants en termes d'efficacité, la gestion des

clés peut poser des problèmes lors de communications entre plusieurs utilisateurs.

C'est en 1976 que W. Diffe et M. Hellman ont proposé une solution à ce problème en introduisant le concept de cryptographie asymétrique ou cryptographie à clé publique. Cependant, il a fallu attendre 1978 et l'invention de Rivest Shamir Adleman (RSA) pour fournir cette nouvelle famille d'algorithmes.

RSA est une méthode de cryptage basée sur l'utilisation de deux nombres premiers très grands. Si les nombres choisis actuellement ont une taille de 4096 bits, il serait normalement extrêmement difficile, voire impossible, de casser un chiffre RSA. Cependant, des choix incorrects, tels que l'utilisation de nombres premiers insuffisamment grands, peuvent limiter l'efficacité de cette méthode. Certaines attaques, telles que celles basées sur les courbes elliptiques, ont montré leur efficacité dans de telles circonstances.

La structure de ce document est la suivante :

Dans le chapitre 1, nous effectuons une revue de l'état de l'art de la cryptographie, le protocole de chiffrement TKIP et la méthode RSA dans le détail, en examinant leur historique, le théorème de Fermat, etc.

Dans le second chapitre, nous nous focaliserons sur les attaques TKIP et les attaques RSA connues, en étudiant leurs impacts et leurs limites.

Le chapitre 3 sera consacré à l'implémentation de l'algorithme RSA et à une attaque du chiffrement TKIP sur un réseau sans fil.

Le chapitre 4 sera réservé aux résultats et à la discussion du chapitre précèdent.

Finalement, ce mémoire se terminera par une conclusion en tenant compte des tests effectués.

CHAPITRE 1

Cryptographie, RSA et TKIP

CRYPTOGRAPHIE

1. Introduction

La sécurité de l'information est une préoccupation majeure dans le contexte actuel, où les échanges de données sensibles sont omniprésents. La cryptographie, en tant qu'outil essentiel de protection des informations, joue un rôle crucial dans la préservation de la confidentialité, de l'intégrité, de l'authentification et de la non-répudiation des données. Ce domaine d'étude s'est développé pour fournir des méthodes et des protocoles permettant de sécuriser les communications et de prévenir les attaques malveillantes [2].

Cryptographie

1.1 Concepts de base

- **Cryptologie** : La cryptologie est la science qui étudie les aspects scientifiques des techniques de la cryptographie et de la cryptanalyse. Elle englobe à la fois la conception des méthodes de chiffrement et de déchiffrement, ainsi que l'analyse de ces méthodes pour en évaluer leur sécurité. [2]

- **Cryptographie** : La cryptographie est l'ensemble des techniques utilisées pour rendre un message inintelligible ou incompréhensible. Elle vise à assurer la confidentialité, l'intégrité et l'authenticité des données. La cryptographie se base sur des algorithmes et des clés de chiffrement pour transformer le texte en clair (plaintext) en un texte chiffré (ciphertext), rendant ainsi le message illisible sans la clé appropriée[6].

- **Chiffrement** : Le chiffrement des données consiste à convertir des données en utilisant une clé secrète ou un mot de passe, de sorte que seules les personnes disposant de cette clé puissent les lire. Le chiffrement se fait en utilisant une clé de chiffrement pour transformer les données en texte chiffré. Les données initiales en texte clair sont ainsi protégées et sécurisées.[7]

- **Déchiffrement** : Le déchiffrement est le processus inverse du chiffrement. Il consiste à retrouver le texte original d'un message chiffré en utilisant la clé de déchiffrement appropriée. Le déchiffrement permet de récupérer les données en texte clair à partir du texte chiffré, rendant ainsi le message compréhensible pour le destinataire autorisé.[8]

- **Cryptanalyse** : La cryptanalyse est l'art de casser des méthodes cryptographiques existantes dans le but de démontrer leurs vulnérabilités ou de récupérer des informations secrètes sans posséder la clé de déchiffrement. Elle implique l'analyse des algorithmes de chiffrement et des messages chiffrés afin de trouver des faiblesses qui pourraient permettre une attaque réussie. [9]

- **Décryptage** : Le décryptage est le processus de récupération du texte original d'un message chiffré sans posséder la clé de déchiffrement. Contrairement au déchiffrement, qui utilise une clé de déchiffrement, le décryptage tente de retrouver le message en utilisant des techniques de cryptanalyse ou des méthodes non autorisées[10].

1.2 Historique

La cryptographie, depuis ses débuts, a été développée dans le but de sécuriser les communications en rendant les informations confidentielles pour les destinataires non autorisés. Étymologiquement, le mot "cryptographie" provient du grec "kryptos", qui signifie "cacher", et "graphein", qui signifie "écrire". Ainsi, la cryptographie peut être définie comme l'étude des différentes méthodes et techniques mathématiques utilisées pour garantir la confidentialité, l'intégrité et l'authenticité des messages.

Son rôle principal est de permettre le stockage ou la transmission d'informations sensibles à travers des réseaux non sécurisés, tels qu'Internet, de manière à ce qu'elles ne puissent être lues ou compromises par des individus non autorisés. La cryptographie concerne la transformation d'un message intelligible (texte, image, chiffres) en un message codé, rendant sa compréhension impossible pour toute personne n'ayant pas le code de déchiffrement approprié.

Il est important de noter que la cryptographie diffère de la stéganographie, qui vise à dissimuler un message sans le transformer de manière "logique". La stéganographie

se concentre sur la dissimulation du message au sein d'un autre support (comme une image) sans que son existence ne soit détectée. [2]

1. Systèmes de chiffrement

Il existe deux systèmes de chiffrement :

1.1 Systèmes de chiffrement à clé secrète, ou systèmes symétriques [2]

La cryptographie symétrique repose sur le partage d'une même clé secrète (S) entre deux interlocuteurs en communication. Cette clé secrète est utilisée pour paramétrer un algorithme de chiffrement et de déchiffrement. Avant toute communication, il est essentiel d'échanger physiquement la clé secrète. Pour le stockage de messages, le principe reste le même, mais avec un seul interlocuteur.

La clé secrète (S) est généralement représentée par un ensemble de bits de taille limitée. L'échange physique de cette clé secrète est crucial pour garantir la sécurité des communications. Cependant, il existe un risque potentiel si la clé est interceptée ou compromise lors de son échange.

Une méthode courante pour tenter de retrouver le contenu des communications est l'« attaque par force brute », qui consiste à essayer toutes les clés possibles. Le nombre de clés possibles dépend de la taille de la clé (S). Pour une clé de n bits, il y a 2n clés possibles. Ainsi, la complexité d'une attaque par force brute est limitée par la taille de cet ensemble de clés.

Il convient de noter que, généralement, la clé secrète commune (S) n'est pas utilisée directement pour chiffrer les messages. Au lieu de cela, elle est utilisée pour chiffrer une autre clé (K) qui est générée aléatoirement par l'émetteur pour chaque session ou message. Cette clé (K) chiffrée est ensuite envoyée en début de session ou de message ou conservée avec le message dans le cas du stockage.

Ce processus de chiffrement symétrique assure la confidentialité des messages, car la connaissance de la clé secrète (S) est nécessaire pour déchiffrer les messages chiffrés. Cependant, le principal défi de la cryptographie symétrique réside dans la gestion sécurisée de la clé secrète (S) et dans la nécessité d'un échange préalable entre les parties..

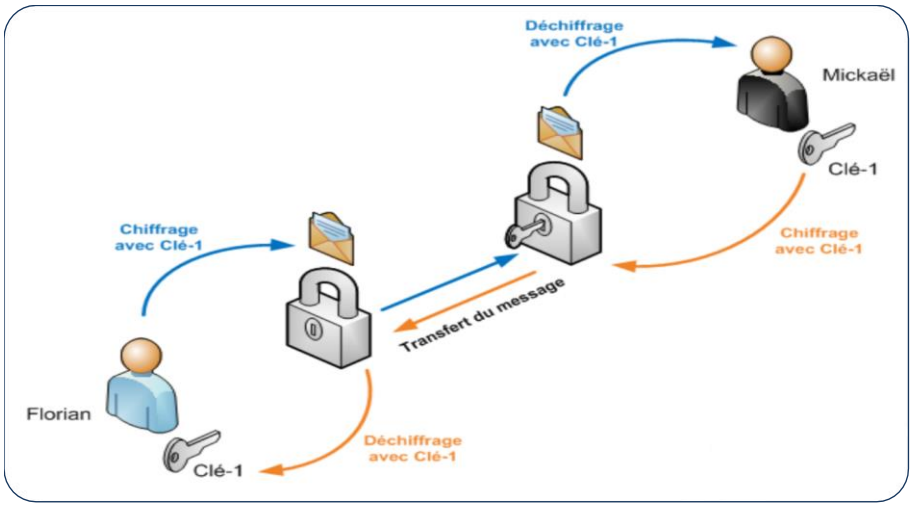


Figure I. 1: Schéma de cryptographie symétrique [2]

❖ **Les algorithmes de chiffrement symétrique**

Les algorithmes de chiffrement symétrique couramment utilisés incluent AES, TKIP, DES et 3DES.

TKIP est fondé sur un algorithme de chiffrement symétrique RC4 (dont les vulnérabilités sont connues depuis 1995) et AES (WPA/WPA2/WPA3). En matière de chiffrement et d'intégrité, le protocole TKIP décrit dans WPA utilise toujours l'algorithme RC4 compatible avec l'ancien matériel. TKIP (Temporal Key Integrity Protocol) est la version corrigée des faiblesses du WEP. AES est considéré comme le plus sécurisé et est utilisé dans de nombreux protocoles de sécurité. DES est plus ancien et moins sécurisé, tandis que 3DES est une version améliorée de DES. [12]

➢ **Avantage**

Les avantages de la cryptographie symétrique sont : son efficacité, sa simplicité et sa résistance à certaines attaques. Cependant, elle présente des limitations telles que la distribution sécurisée des clés et le manque d'évolutivité.

➢ **Utilisation**

La cryptographie symétrique est utilisée dans divers domaines, tels que les communications sécurisées, le stockage sécurisé, l'authentification et le contrôle

d'accès. Elle est largement utilisée pour protéger les données et garantir la confidentialité, l'intégrité et l'authenticité des informations.

1.2 Systèmes de chiffrement a clé public ou systèmes asymétriques

Les crypto-systèmes asymétriques, également appelés crypto-systèmes à clé publique, a été inventés dans les années 1970 par Whitfield Diffie et Martin Hellman. Ils ont présentés leur invention dans un article intitulé "New Directions in Cryptography" en1976. Leur idée était de résoudre un problème fondamental dans la cryptographie : comment deux parties peuvent-elles communiquer de manière sécurisée sans avoir besoin de se mettre d'accord sur une clé secrète à l'avance ?

Les crypto-systèmes asymétriques résolvent ce problème en utilisant des paires de clés différentes pour chiffrer et déchiffrer les données. L'une de ces clés, la clé publique, est disponible pour tout le monde, tandis que l'autre, la clé privée, est connueuniquement de la partie qui l'a émise. La clé publique peut être utilisée pour chiffrer lesdonnées, mais elle ne peut pas être utilisée pour les déchiffrer. Seule la clé privée correspondante peut être utilisée pour déchiffrer les données. Contrairement aux crypto-systèmes symétriques qui utilisent une seule clé pour chiffrer et déchiffrer les données [1].

- **Fonctionnement des crypto-systèmes asymétriques :**

Le fonctionnement des crypto-systèmes asymétriques repose sur des complexes mathématiques, notamment l'arithmétique modulaire et la théorie des nombres. Il comprend une série d'étapes bien définie et chacun d'eux garantit le bon fonctionnement du système. Les algorithmes à clé publique ont une double fonction : d'une part, ils servent à chiffrer des messages, et d'autre part, ils permettent de calculer des signatures numériques. Une signature numérique est une valeur qui dépend à la fois du message, considérée sous sa forme numérisée comme un nombre, et de l'identité du signataire, qui doit être la seule personne capable de la calculer. Ainsi, un message signé est, constitué du message en clair ainsi que de cette signature numérique. Pour vérifier une signature, il suffit d'appliquer la fonction inverse de la

signature afin de retrouver le message original en clair [2].

Chaque utilisateur possède un couple de clés différent, composé de la clé S gardée secrète par le propriétaire, utilisée pour déchiffrer des messages reçus ou signer des messages, et de la clé P rendue publique. La clé **P** dépend de la clé **S** par une fonction à sens unique facilement calculable, mais son inversion est extrêmement difficile, ce qui signifie qu'il est impossible de déduire la clé **S** à partir de la clé publique **P**. La clé publique **P** est utilisée par n'importe qui pour chiffrer des messages destinés au propriétaire des clés ou pour vérifier les signatures. Lorsqu'un correspondant **B** souhaite chiffrer un message destiné à la personne **A**, il applique la fonction définie par la clé publique $\mathbf{P_A}$ de **A**. **A** peut alors déchiffrer le message à l'aide de sa clé secrète$\mathbf{S_A}$, qui est connue uniquement d'elle-même. Pour signer un message, **B** applique la fonction définie par sa clé secrète $\mathbf{S_B}$ pour calculer une signature unique. Pour vérifier cette signature, **A** lui applique la fonction inverse de la fonction de signature, définie par la clé publique $\mathbf{P_B}$ de **B**, ce qui lui permet de retrouver le message initial en clair. Il est important de souligner que seul B, qui détient la clé secrète $\mathbf{S_B}$, est capable de calculer cette signature [2].

B envoie un message chiffré et/ou signé à A	
B chiffre avec $\mathbf{P_A}$	**A** déchiffre avec $\mathbf{S_A}$
B signe avec $\mathbf{S_B}$	**A** vérifie avec $\mathbf{P_B}$

Tableau I. 1: envoie de message signé ou non

Compte tenu de leur lenteur, les algorithmes à clé publique ne sont généralement pas utilisés pour chiffrer des messages directement. Au lieu de cela, des algorithmes à clé symétrique sont employés pour cette tâche. Toutefois, pour garantir la sécurité des échanges, la clé de session aléatoire générée pour le chiffrement symétrique est elle-même chiffrée à l'aide de l'algorithme à clé publique. Ce dispositif permet de combiner les avantages de chaque méthode, tout en limitant les inconvénients. Ce procédé est similaire à celui des systèmes à clé secrète.

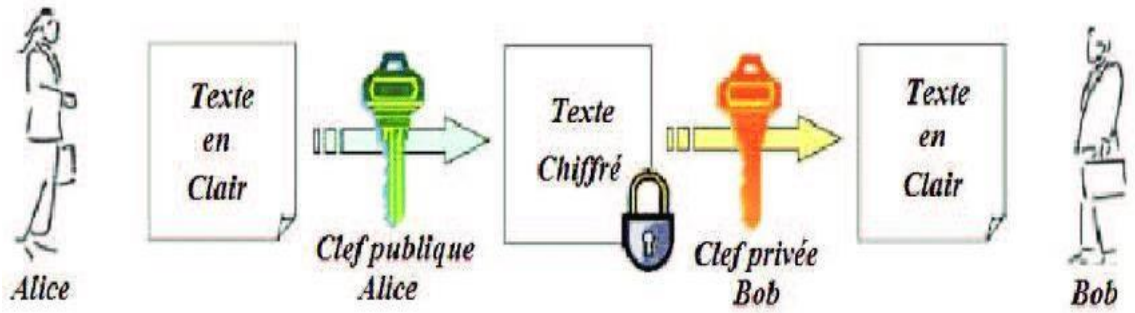


Figure I. 2 : Schéma de cryptographie asymétrique

- **Utilisation** :

Les crypto-systèmes asymétriques sont largement utilisés pour la sécurité des communications sur Internet. Le protocole SSL/TLS, qui est utilisé pour sécuriser les connexions HTTPS entre les navigateurs web et les serveurs web, utilise des crypto-systèmes asymétriques pour échanger les clés de session qui sont utilisés pour chiffrer les données.

- **Avantages du chiffrement à clé publique**

- Pas de problèmes de gestion de clés.

- Chaque interlocuteur ne garde secret que sa clé privée [3].

- **Inconvénients du chiffrement à clé publique**

- Les algorithmes asymétriques sont particulièrement lents à comparer avec ceux symétriques.
- Utilise plus de ressources

- Ils nécessitent des clés plus longues (1024 bits, 2048 bits) [3].

Finalement comme nous avons vu précédemment, les deux systèmes de bas de la cryptographie souffrent de problème complémentaire. Ainsi pour renforcer la sécurité des systèmes de chiffrement réside certainement dans l'utilisation combinée de ces deux techniques, ce qui est appelé la cryptographie hybride ou mixte. [11]

Protocole TKIP

Le protocole de chiffrement TKIP (Temporal Key Integrity Protocol) est un protocole de sécurité utilisé dans les réseaux sans fil pour protéger les communications. Il a été développé comme une amélioration de WEP (Wired Equivalent Privacy) pour remédier à ses vulnérabilités. [13]

1. **Histoire**

   TKIP a été introduit en 2003 comme partie de la norme de sécurité sans fil WPA (Wi-Fi Protected Access), qui était une alternative améliorée à WEP. TKIP était destiné à fournir un niveau de sécurité plus élevé que WEP en utilisant des clés dynamiques et des techniques de chiffrement plus robustes. [14]

2. **Objectif**

   TKIP visait à renforcer la sécurité des réseaux sans fil en résolvant les vulnérabilités connues de WEP, telles que l'utilisation de clés statiques, l'absence d'intégrité des données et la faiblesse des mécanismes de chiffrement.

3. **Mécanismes de chiffrement TKIP :**

   Tout comme le WEP, TKIP repose sur l'algorithme de chiffrement RC4, mais il a été conçu pour permettre aux systèmes basés sur le WEP de bénéficier d'un protocole plus sécurisé. TKIP est essentiel pour la certification WPA et est inclus en option dans le RSN 802.11i. Il apporte des corrections pour chaque vulnérabilité du WEP.[14]

   Le mécanisme de chiffrement TKIP (Temporal Key Integrity Protocol) pour la trame de la couche MAC (Media Access Control) utilise les étapes suivantes pour traiter les MSDU (MAC Service Data Unit) : **[15]**

   - Éviter les clés faibles : Utilisation de clés de chiffrement de taille suffisamment grande pour éviter les vulnérabilités liées aux clés faibles.
   - IV de 16 bits (partie inférieure) et IV de 32 bits (partie supérieure) : Génération d'un nombre aléatoire pour chaque MPDU (MAC Protocol Data

Unit) afin de créer un IV unique.

- Michael (TMK + SA + priorité + texte clair MSDU) : Calcul d'un code d'intégrité de message (MIC) sur la trame en utilisant une clé MIC dérivée de la clé principale, qui comprend le Traffic Encryption Key (TMK), l'adresse de destination (SA), la priorité et le texte clair MSDU.

- ICV=CRC (texte clair || MIC) : Utilisation d'un mécanisme de contrôle d'intégrité (ICV) basé sur le code de redondance cyclique (CRC) pour assurer l'intégrité des données, en concaténant le texte clair avec le MIC.
- TTAC (TKIP – mélange d'adresse de transmission et de clé) - 80 bits : Utilisation d'un mélange de clés pour renforcer la sécurité, en combinant l'adresse de transmission avec une partie de la clé.

- Génération de keystream : Création d'un keystream (séquence de bits générée à partir de la clé de chiffrement) pour le chiffrement des données.

À l'arrivée d'une MPDU :

- Extraction de l'IV : Récupération de l'IV pour chaque MPDU afin de générer la clé de paquet.

- Vérification du numéro de séquence : Vérification de la validité du numéro de séquence pour détecter d'éventuelles altérations.

- Génération de la clé de paquet : Utilisation de l'IV et de la clé principale pour générer la clé de déchiffrement du paquet.

- Déchiffrement du paquet : Déchiffrement du contenu de la MPDU en utilisant la clé de paquet appropriée.

- Regroupement des MPDU : Rassemblement des MPDU correspondant à une même MSDU.

- ❖ Calcul du code MIC : Calcul du code MIC à partir du paquet déchiffré et comparaison avec celui contenu dans le message. Si les résultats diffèrent, la MSDU est rejetée.

- ❖ Remontée de l'MSDU : L'MSDU déchiffrée et vérifiée est ensuite transmise aux couches supérieures pour traitement ultérieur.

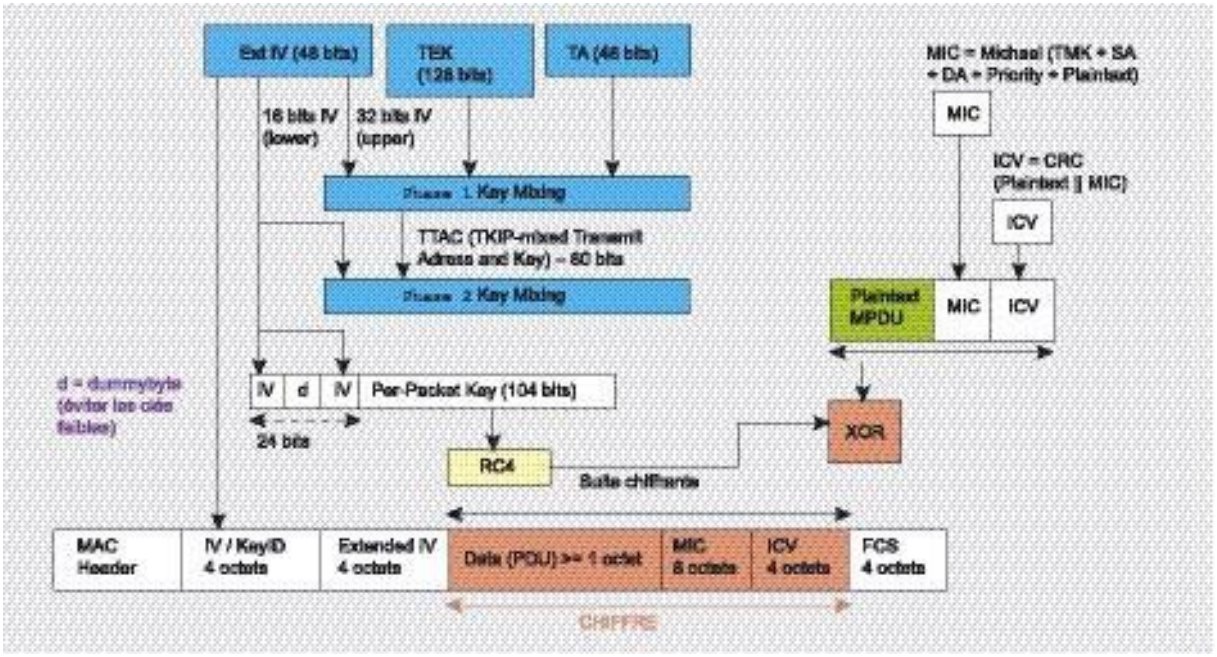


Figure I. 3 : *Schéma TKIP de mixage des clés et de chiffrement [14]*

4. Mise à jour du chiffrement

TKIP inclut également des mécanismes pour la mise à jour périodique des clés de chiffrement afin de réduire les risques de compromission à long terme. Cela permet de renforcer la sécurité et de rendre plus difficile le déchiffrement des données passées.

5. Remplacement par WPA2

Bien que TKIP ait amélioré la sécurité par rapport à WEP, il a été remplacé ultérieurement par WPA2 (Wi-Fi Protected Access 2).

WPA2 utilise le protocole de chiffrement AES (Advanced Encryption Standard) qui est considéré comme plus robuste et sécurisé que TKIP.

Il existe des attaques connues contre le chiffrement TKIP, telles que l'attaque par dictionnaire et l'attaque par force brute, qui exploitent les faiblesses du protocole. Cependant, ces attaques peuvent être contrées par des mesures de sécurité appropriées et l'utilisation de protocoles plus récents.

RSA

Le chiffrement symétrique existe depuis l'aube de la cryptographie.il ne fut remis en question qu'en 1976 par Diffie et Hellman. Après la publication de leur article qui a introduit l'idée du chiffrement à clé publique (également appelé chiffrement asymétrique) trois chercheurs nommé Ron Rivest (cryptographe), Adi Shamir (expert en cryptanalyse) et Len Adleman (Chercheurs en Informatique et Biologie Moléculaire)ont décidé de travailler ensemble sur Le nouveau système de codage révolutionnaire de W. Diffie et M. Hellman que c'est un fait logiquement impossible (il présentait des failles). Ils ont essayé de prouver que le crypto-système asymétrique ne peut exister. Mais après de nombreuses tentatives infructueuses, ils ont trouvé le contraire, un candidat possible de crypto-système asymétrique connu aujourd'hui sous le nom de RSA [3].

Figure I. 4: Les trois chercheurs

1. Historique :

L'algorithme RSA a révolutionné le monde de la cryptographie en introduisant pour la première fois des méthodes de cryptographie à clé publique. Il a été inventé en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman, et est basé sur la difficulté de factoriser de grands nombres premiers. Le principe de la cryptographie à clé publiquea été introduit à l'état de concept par Whitfield Diffie et Martin Hellman, et

l'algorithme RSA est aujourd'hui le crypto-système asymétrique le plus connu et le plus utilisé dans le monde entier. En raison de sa popularité, l'algorithme RSA est l'un des algorithmes les plus étudiés,tant d'un point de vue théorique que pratique. De nombreux experts en sécurité le recommandent comme un moyen efficace de protéger les données sensibles. Cependant, il est important de noter que même les meilleurs algorithmes de cryptographie ne sont pas invulnérables, et il est toujours possible qu'ils soient compromis si des attaquants sont en mesure de trouver des vulnérabilités ou des failles dans leur mise en œuvre.

2. Définition et applications

RSA est un algorithme cryptographique qui fait partie de la catégorie des algorithmesà clé publiques ou systèmes asymétriques. L'acronyme RSA, vient du nom de ses troisinventeurs (Rivest, Shamir et Adleman). Ce chiffrement est le plus connu en termes d'algorithmes cryptographiques de type asymétrique [3].

L'importance de RSA se reflète dans son utilisation dans de nombreuses applications, allant des transactions sécurisées sur Internet aux cartes à puce bancaires et aux systèmes de sécurité gouvernementaux. RSA est un système universel et a été intégré dans de nombreux standards informatiques. Il est également utilisé dans les cartes Ethernet, les clés USB sécurisées et de nombreuses institutions gouvernementales, militaires et universitaires.

3. Définition de l'algorithme

L'algorithme **RSA** est utilisé pour la cryptographie à clé publique et est basé sur le fait qu'il est facile de multiplier deux grands nombres premiers mais difficile de factoriser le produit lorsqu'on a des ressources limités. C'est l'exemple le plus courant de cryptographie asymétrique, toujours considéré comme sûr, avec la technologie actuelle, pour des clés suffisamment grosses (***1024, 2048*** *voire* ***4096 bits***) [2].

4. Principe de Fonctionnement du RSA

4.1 Génération des clés [4]

➢ Etape1 :

- On génère deux grands nombres premiers p et q (doit être absolument garder secret et doit être le plus grand possible afin de compliquer au maximum la tâche à celui qui veut attaquer le système.)

Soit N = p*q, N est appelé module de chiffrement.
Soit m = (p-1) * (q-1), m est appelé l'indicatrice d'Euler

➢ Etape2 :

- On Choisit un nombre e (appelé exposant de chiffrement d'où le choix de lalettre e pour encryption) premier avec m (choix fréquent : e = 3)
- e étant premier avec m, il est, d'après le théorème de Bachet-Bézout, inversible mod m, c'est-à-dire qu'il existe un entier d tel que e*d ≡ 1 mod m où
  d est l'exposant de déchiffrement (d'où le choix de la lettre d pour décryption).
- Clés obtenues :

Clé publique : (e, N) Clé privée : (d, N)

4.2 Chiffrement et Déchiffrement

- Un message m est chiffré en $C = m^e \ mod \ N$, où e est l'exposant de chiffrementet C le message chiffré.

● Le déchiffrement d'un chiffré C n'est autre que : $C^d \ mod \ N$, tel que d est l'exposantde déchiffrement.

Pour chiffrer un message, il suffit de connaitre e et N. D'autre part pour déchiffrer, ilfaut d et N [3].

4.3 Vérification

La vérification de la signature RSA étant une technique permettant de vérifier l'authenticité, l'intégrité et la non-répudiation des données

signées à l'aide de la technique de signature numérique RSA. Elle utilise aussi la clé publique de l'émetteurpour déchiffrer la signature et comparer le résultat au hachage des données reçues. Si le résultat est identique, cela prouve que les données n'ont pas été altérées pendant la transmission et que l'émetteur ne peut pas nier avoir envoyé les données

5. Algorithmes de base pour le RSA [3]

Algorithme 1. Génération de clés avec RSA.

Entrée : taille de la clé //exprimée en bits.
Sortie : la clé publique (N, e) et la clé privée (N, d)

1. Prendre deux nombres premiers **p** et **q** suffisamment grands (de taille à peu près égale).
2. Calculer N = p.q,
3. Calculer $\varphi(N)$= (p-1) × (q-1),
4. Choisir un nombre e tel que $1 < e < \varphi(N)$ et le $pgcd\left(e, \varphi(N)\right) = 1$,
5. Prendre un nombre e qui n'a aucun facteur en commun avec $\varphi(N)$,
6. Calculer d tel que $ed \bmod \varphi(N)$=1.
7. **Return** N, e, d.

Algorithme 2. Chiffrement avec RSA.

Entrée : (N, e) et m // la clé publique et le texte en clair m avec $m \in$ [0.N-1].
Sortie : C // texte chiffré.

1. Calculer **C**=m^e **mod N**
2. **Return** C

Algorithme 3. Déchiffrement avec RSA.

Entrée : (d, e) et C // la clé privée et le texte chiffré.
Sortie : m // texte clair.

1. Calculer $m = C^d$ **mod N**
2. **Return** m

Algorithme 4. Signature avec RSA.

Entrée : (N,d) et m // la clé privée et le texte en clair m avec $m \in [0.N-1]$.
Sortie : S, m // la signature et le texte clair.
1. Calculer h=H(m) // le hachage du message m
2. Calculer S= h^d mod N
3. **Return** S, m

Algorithme 5. Vérification de la signature.

Entrée : (e, N), m,S
Sortie : Acceptation ou rejet de la signature
1. Calculer h=H(m)
2. Calculer h'=S^emod N
3. Accepter si h=h', rejet sinon

6. EXEMPLE :

Alice choisit deux nombres premiers :

p = 11 et q = 19 alors $N = pq$

implique que N = 209.m = 10 × 18 =

180.

Le choix d'Alice tombe par exemple sur e=7 premier

avec m = 180.La clé publique est (7, 209).

Alice envoie la clé publique (7, 209) à Bob.

Alice calcule maintenant d tel que le reste de 7d, dans la division par m = 180, soit 1. Elle trouve 103 (on vérifie que 7×103=721 a bien pour reste 1 dans la division par 180).La clé secrète est 103.

Pour trouver cette clé secrète, Alice a eu besoin de m = (11 − 1)(19 − 1) et que les deux nombres premiers 11 et 19 ne sont connus que par elle seule.

Bob choisit le nombre 63, le message qui doit être chiffré.

Il a reçu la clé publique (7,209) et doit calculer c^e $mod\ N$

soit 637 mod 209.Il trouve 123 et envoie cette valeur à

Alice.

Alice reçoit 123. Elle utilise sa clé secrète d = 103 pour calculer 123^d $mod\ N$ soit123^{103} mod 209.

Elle trouve 63.

Elle a bien trouvé le nombre choisi par Bob.

Ce calcul a nécessité l'utilisation de deux facteurs p et q choisis par Alice et inconnusde Bob.

P, Q sont indispensables pour trouver la clé secrète.

Et c'est justement la détermination de ces deux-là qui pose une réelle difficultélorsqu'elles sont très grandes.

7. Utilisation de l'algorithme RSA

- Pour garantir une sécurité adéquate lors du chiffrement RSA, il estessentiel de respecter plusieurs règles :

- Éviter le chiffrement de blocs trop courts, qui peuvent être vulnérables à des attaques.
- Utiliser des nombres premiers de grande taille pour p et q, afin que leur produit (p-1) x (q-1) soit également grand.
- Utiliser des valeurs de N très grandes pour compliquer la tâche des attaquants.
- Éviter l'utilisation de la même valeur de N pour plusieurs clés RSA, car cela peut faciliter les attaques en partageant des informations entre lesclés.
- Si une paire (N, d) est compromise, il est important de ne plus l'utiliser pour la cryptographie RSA.

8. Conclusion

Dans ce chapitre nous avons abordé la cryptographie et ses deux grandes parties : symétrique qui est l'algorithme de chiffrement qui utilise une seule clé pour le chiffrement et déchiffrement et asymétrique qui est l'algorithme de chiffrement qui utilise une paire de clés : une clé publique et une clé privée. La clé publique peut être distribuée

librement, tandis que la clé privée est gardée secrète. Ces deux clés sont mathématiquement liées, de sorte que les données chiffrées avec la clé publique ne peuvent être déchiffrées qu'avec la clé privée correspondante, et vice versa. On a aussi mis l'accent sur TKIP qui est l'un des protocoles de chiffrement symétrique pour les réseaux sans fil et RSA qui est l'un des crypto-systèmes asymétriques les plus connus et les plus utilisés, en détaillant son principe de fonctionnement, à sa cryptanalyse et son utilisation. Dans ce qui suit nous allons présenter leurs attaques les plus connus, avec leurs impacts et leurs limites.

CHAPITRE2

Les Attaques RSA et TKIP

1. Introduction

RSA est l'un des algorithmes de cryptographie asymétrique les plus largement utilisés pour sécuriser les communications et les transactions en ligne. Les composants de sécurité sont souvent ciblés par des attaques car ils contiennent des informations confidentielles. Ces attaques cherchent à compromettre la confidentialité,l'intégrité ou l'authenticité des données protégées par les algorithmes de cryptographie embarqués sur ces composants. Ils sont divisés en trois 3 catégories selon la faille qu'elle exploit, nous avons : les attaques physiques également, connues sous le nom d'attaques matérielles, exploitent les vulnérabilités liées à l'implémentation matérielle de ces algorithmes, les attaques mathématiques, sont des attaques qui exploitent des faiblesses dans les fondements mathématiques de l'algorithme cryptographique et enfin les attaques de protocoles, qui vise à exploiter une faiblesse dans la manière dont les protocoles d'utilisation sont mis en œuvre pour compromettre la sécurité de l'algorithme [3].

2. Définition d'une attaque

Une « attaque » est le fait d'exploiter une faille d'un système informatique (systèmed'exploitation, logiciel) à des fins non connues par l'exploitant du système et généralement préjudiciables [5].

3. Les catégories d'attaques RSA

a. Attaques physiques

❖ **Attaque sur le temps de calcul**

L'algorithme de calcul de la fonction RSA utilise une boucle pour le calcul de y^d $\boldsymbol{mod}$ N, où $\boldsymbol{d}$ est la clé secrète et est représenté par $\boldsymbol{d} = \boldsymbol{d1}, \boldsymbol{2},\boldsymbol{3}, \ldots, \boldsymbol{dn}$, où chaque bit $\boldsymbol{di}$ est égal à 0 ou 1. Cependant, cette boucle de calcul peut créer des instructions qui, selon la valeur de $\boldsymbol{di}$, ne seront pas toutes exécutées, ce qui peut créer des différences de temps de calcul mesurables. Cette vulnérabilité permet aux attaquants de retrouver la clé secrète d bit par bit en mesurant les temps de calcul

pour de nombreuses valeurs initiales de y (message crypté). Afin de prévenir ce type d'attaque, des techniques de masquage peuvent être utilisées pour rendre les différences de temps de calcul indépendantes de la valeur de $\boldsymbol{di}$ [2].

- **Attaque sur la consommation électrique**

  L'attaque par analyse de la consommation électrique est une méthode d'attaque qui consiste à mesurer la consommation électrique d'un dispositif lors de l'exécution d'un algorithme cryptographique afin de retrouver des informations sensibles telles que la clé secrète. Cette méthode est basée sur le fait que la consommation électrique varie en fonction des opérations effectuées par le dispositif. En analysant la courbe de consommation électrique, il est possible de déduire certaines informations sur les opérations effectuées, telles que la valeur de la clé secrète d en utilisant l'algorithme de chiffrement RSA. Cependant, il existe des techniques de contre-mesures qui permettent de réduire l'impact de cette attaque, telles que l'ajout de bruit aléatoire à la consommation électrique afin de rendre plus difficile la déduction d'informations sensibles [2]**.**

- **Attaque par injection de faute**

Les attaques par faute sont une méthode d'attaque qui consiste à introduire des erreurs dans le crypto-système afin de provoquer un comportement non recherchédes opérations cryptographiques et d'extraire des informations secrètes telles qu'une clé de chiffrement. Ces attaques peuvent être réalisées sur des composants matériels ou logiciels. En plus des attaques sur la consommation électrique et le temps de calcul,les attaques par faute peuvent être combinées avec d'autres attaques pour augmenter les chances de succès. Pour se protéger contre ces attaques, il est important de mettre en place des mécanismes de détection et de correction d'erreurs, ainsi que des méthodes de protection de l'état interne du système, telles que la duplication de circuits et la redondance [3]

b. Attaques mathématiques

❖ Attaque de Wiener

Proposée par le cryptologue Michael J. Wiener. Cette attaque permet de retrouver facilement la clé privée à partir de la clé publique (e, N), lorsque les conditions (d trop petit) et q<p<2q (p et q trop proches) sont remplies, il est facile de retrouver *d*. Cette attaque a été améliorée par Dan Boneh et Glen Durfee pour tous les exposants d inférieurs ou égaux à $N^{0.292}$ [3].

❖ **Factorisation de modulo N**

La factorisation de modulo N est une méthode utilisée pour factoriser un entierN en deux facteurs premiers. Elle est basée sur le théorème de Fermat qui affirme que si p est un nombre premier et a est un entier naturel non divisible par p, alors a^ (p-1) - 1 est divisible par p.

La méthode consiste à trouver un entier a qui est choisi au hasard entre 2 et

N-1, et à calculer le PGCD de N avec (a^(p-1) - 1) où p est un multiple de tous les facteurs premiers de N. Si le PGCD est différent de 1 et de N, alors on a trouvé un facteur de N.

Le choix de l'entier a est crucial pour la réussite de la méthode. En effet, si a est choisi tel que PGCD (N, a) est différent de 1, alors la méthode ne fonctionnera pas. Dans ce cas, il faut choisir un autre entier a et répéter le processus jusqu'à ce qu'un facteur de N soit trouvé.

La méthode de factorisation de modulo N est lente et peu efficace pour les grands entiers N. Cependant, elle est utilisée dans certaines attaques cryptographiques, notamment dans l'attaque de Fermat, qui est une attaque parfactorisation du RSA. Les chercheurs en sécurité cherchent constamment de nouvelles méthodes pour améliorer la factorisation des grands nombres premiers, car de nombreux protocoles de sécurité dépendent de la difficulté à factoriser des grands nombres premiers [3].

❖ **Attaque en connaissant quelques bits de la clé privée**

Une attaque en connaissant quelques bits de la clé privée est une technique d'attaquequi vise à récupérer la clé privée d'un système de

cryptographie asymétrique en utilisant des informations partielles sur celle-ci. Plus précisément, l'attaquant dispose de quelques bits de la clé privée et utilise ces informations pour déduire les autres bits.

D. Boneh et al ont montré que si la taille de la clé d est de k bits alors la connaissance de $\frac{k}{4}$ bits de poids faible est largement suffisante pour récupérer la clé d [3].

c. Attaque de protocole

Bien que RSA soit considéré comme un algorithme robuste, l'utilisation qui en est faite peut comporter des risques. Par exemple, si un même message est chiffré avec la clé publique RSA de trois destinataires différents, il est possible de retrouver le message original en combinant les trois messages chiffrés en utilisant la propriété de multiplicativité de la fonction RSA : f(x*y) = f(x) × f(y). Demême, il est risqué d'utiliser la même clé publique RSA pour chiffrer plusieurs messages liés [3].

❖ **Attaque d'Hastad :**

L'attaque de Hastad, également connue sous le nom d'attaque de Broadcast, est une méthode d'attaque qui s'applique lorsqu'un même message, ou des messages liés par une relation connue, sont chiffrés et envoyés à plusieurs personnes. Pour illustrer cette attaque, supposons que Bob utilise le même exposant de chiffrement e pour envoyer le même message m à k personnes différentes, chacune avec sa propre clé publique (Ni, e). Le message m est inférieur à tous les Ni et Bob chiffre naïvement le message m avec $Ci = me \ mod$ Ni. L'attaquant Marvin peut espionner la communication et obtenir chacun des k messages chiffrés.

Pour simplifier, supposons que l'exposant public e est égal à 3. Dans ce cas, Marvin peut facilement retrouver m si le nombre de messages chiffrés k est supérieur ou égal à 3. En effet, s'il obtient $C1$, $C2$ et $C3$ où $C1 = m^3 \ mod \ N1$, $C2 = m^3 \ mod \ N2$ et $C3 = m^3 \ mod \ N3$, il peut utiliser le théorème des restes chinois

(CRT) pour trouver un entier C' compris entre 0 et $N1.\ N2.\ N3$ tel que $C' = m^3\ mod\ N1.\ N2.\ N3$. Comme le message m est inférieur à chacun des modules Ni, on a $m^3 < N1.\ N2.\ N3$, ce qui permet à Marvin de retrouver m en calculant (dans $\mathbb{Z}$) laracine cubique de C'. Cette attaque peut également être appliquée à n'importe quel exposant de chiffrement e, tant que le nombre de messages chiffrés k est supérieur ou égal à e. Il est donc important d'utiliser des méthodes de chiffrement plus sécurisées pour éviter ce type d'attaque [3].

4. Impacts des attaques RSA

Les attaques RSA peuvent avoir un impact significatif sur la sécurité descommunications qui utilise cet algorithme de chiffrement. Si un attaquant a réussi à casser la clé privée RSA, il peut déchiffrer tous les messages chiffrés avec la clé publique correspondante, ce qui peut porter atteinte à la confidentialité des données sensibles.

De plus, l'attaque de Hastad, qui est une attaque par canal auxiliaire, peut être utilisée pour récupérer le message en clair lorsque le même message est chiffré avec plusieurs clés publiques différentes. Cela peut être particulièrement dangereux si des messages sensibles sont envoyés à plusieurs destinataires à la fois.

En général, les attaques contre RSA peuvent affaiblir la confiance dans la sécurité des systèmes de cryptographie à clé publique, ce qui peut avoir des répercussions sur les secteurs tels que les services financiers, les systèmes depaiement en ligne et les communications sécurisées. Il est donc important de prendre des mesures pour renforcer la sécurité de l'algorithme RSA et des systèmes qui l'utilisent.

5. Limite des attaques RSA

Les attaques RSA ont des limites qui sont principalement dues à la taille desclés utilisées. Les attaques de force brute sont

pratiquement impossibles car la complexité de l'algorithme de factorisation augmente de façon exponentielle avec la taille de la clé. Les attaques par faute et par canal auxiliaire nécessitent un accès physique à l'appareil et sont donc difficiles à réaliser. Les attaques par analyse de temps et par analyse de consommation d'énergie nécessitent des équipements spécialisés et sont également difficiles à réaliser. Enfin, les attaques en connaissant quelques bits de la clé privée ou les attaques par oracle nécessitent un accès au système ou un comportement particulier de celui-ci, ce qui les rend également difficiles à mettre en œuvre. Cependant, la sécurité de RSA dépend également de la qualité de l'implémentation et de la gestion des clés, ce qui peut introduire des vulnérabilités potentielles**.**

Attaque TKIP

Le protocole TKIP (Temporal Key Integrity Protocol) a été développé pour améliorer la sécurité du chiffrement des données dans les réseaux sans fil basés sur la norme IEEE 802.11. Cependant, malgré les améliorations apportées par TKIP par rapport à son prédécesseur WEP (Wired Equivalent Privacy), il a été soumis à plusieurs attaques au fil du temps.

Voici quelques-unes :

1. Les attaques TKIP

1.1 Attaque de réinitialisation de paquets (Packet Re-Inject)

Cette attaque consiste à capturer et réinjecter des paquets chiffrés dans le réseau. Cela peut permettre à un attaquant de réutiliser des paquets chiffrés précédemment envoyés pour obtenir des informations sensibles, telles que des mots de passe ou des clés de chiffrement.

1.2 Attaque de fragmentation (Fragmentation Attack)

Les attaquants peuvent exploiter les vulnérabilités du processus de fragmentation dans TKIP pour forcer la réutilisation de clés de chiffrement et obtenir ainsi l'accès à des données cryptées.

1.3 Attaque par dictionnaire (Dictionary Attack)

Cette attaque vise à deviner la clé de chiffrement en essayant différentes combinaisons de mots de passe possibles à partir d'un dictionnaire préétabli. Si la clé de chiffrement est faible et facilement devinable, l'attaque peut réussir.

1.4 Attaque de réinitialisation de compteur (Countermeasures Reset)

TKIP a été conçu pour mettre en œuvre des contre-mesures pour se protéger contre certaines attaques. Cependant, les attaquants peuvent exploiter les vulnérabilités de ces contre-mesures en réinitialisant les compteurs de TKIP pour continuer leurs attaques.

1.5 Attaque Beck-Tews (Beck-Tews Attack)

Cette attaque exploite des faiblesses dans le protocole TKIP pour récupérer des fragments de texte en clair, ce qui peut conduire à la récupération de la clé de chiffrement.

2. Impact et Limites

- Ces attaques ont sérieusement compromis la sécurité de TKIP, rendant ainsi les réseaux sans fil vulnérables à des attaques plus sophistiquées.
- TKIP est considéré comme obsolète et n'est plus recommandé pour la sécurité des réseaux sans fil. Il a été remplacé par le protocole WPA2 (Wi-Fi Protected Access 2) qui utilise le chiffrement AES (Advanced Encryption Standard) pour une meilleure sécurité.
- Les attaques sur TKIP ont mis en évidence l'importance de mettre en place des protocoles de chiffrement plus solides et de maintenir les normes de sécurité à jour pour faire face aux nouvelles menaces.
- Les attaques sur TKIP ont également souligné l'importance de l'utilisation de mots de passe forts et de clés de chiffrement robustes pour renforcer la sécurité des réseaux sans fil.

CHAPITRE3

Implémentation et conception

1. Introduction

Ce chapitre est entièrement dédié aux aspects pratiques de la conception et de la mise en œuvre de notre application intitulée "Chiffrement et déchiffrement d'un fichier avec l'algorithme RSA" ainsi que de l'attaque du protocole de sécurité des réseaux Wi-Fi WEP/WPA.

2. Ressources utilisées

2.1 Ressources matérielles

- Processeur **Intel(R) Celeron(R) N4000 CPU** @ d'une fréquence de **1.10GHz**
- Processeur **AMD A10-5750M** d'une fréquence de **2.5GHz**
- Une mémoire vive d'une capacité de **4Go**
- Système d'exploitation **64 bits**
- Une carte graphique de **512 Mo**
- Modems **ADSL D-Link**

2.2 Ressources logicielles

- Kali-linux
- Aircrack-ng
- Crunch
- Wireshark
- Python**3.11**
- Pycharm

3. Présentation de l'application python et du craquage d'un réseau wifi

3.1 Application python de l'algorithme RSA

3.1.1 Conception :

Le but principal de notre application est de créer une interface permettant à l'utilisateur de sélectionner un fichier, de le chiffrer et le déchiffrer à l'aide du chiffrement et déchiffrement RSA en utilisant des clés générées dynamiquement, puis créer un fichier chiffré contenant le message chiffré et un fichier déchiffrer contenant le message

déchiffré, et nous permettra aussi de voir le temps pris par chaque fichier chiffrer pour être déchiffrer.

3.1.2 Description de l'application

La structure de notre application est la suivante :

a. Installation de kali-linux

b. Vérification de la version de python, si elle n'existe pas l'installer avec la commande << sudo apt install python3>>

c. Installation de pycharm sur leur site officiel
<< https://www.jetbrains.com/pycharm/download/?section=windows >>

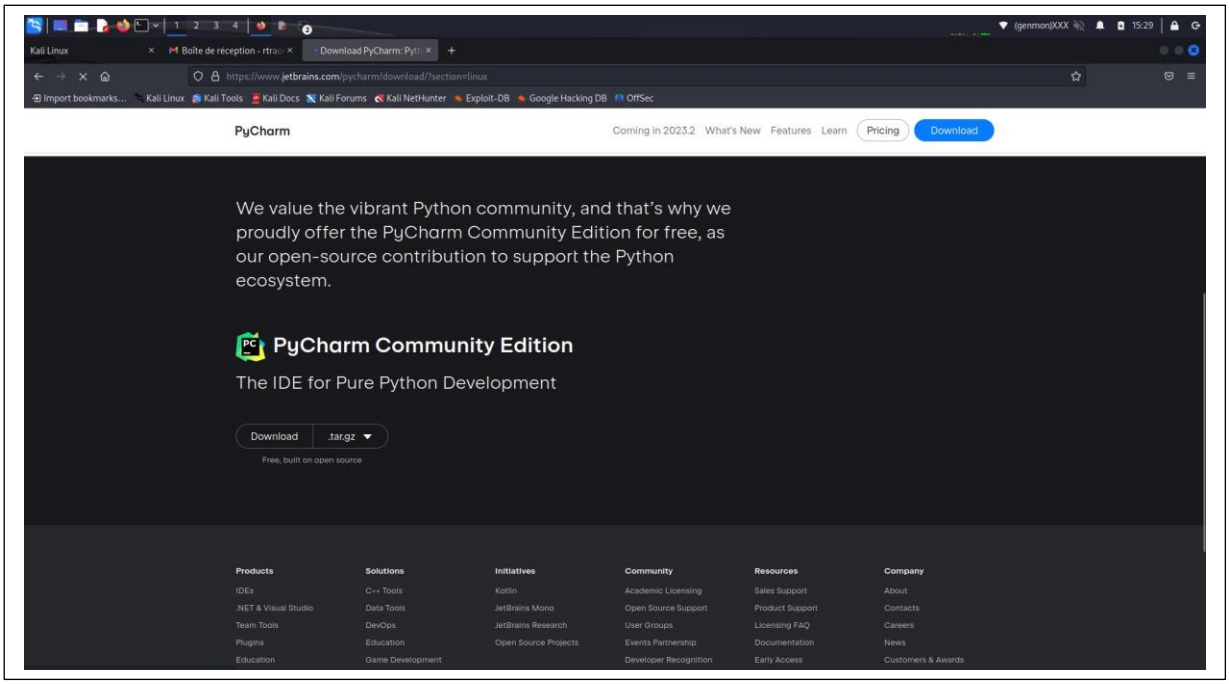


Figure III.1: Téléchargement du zip pycharm

d. Apres avoir télécharger le zip pycharm on décompresse celui-ci

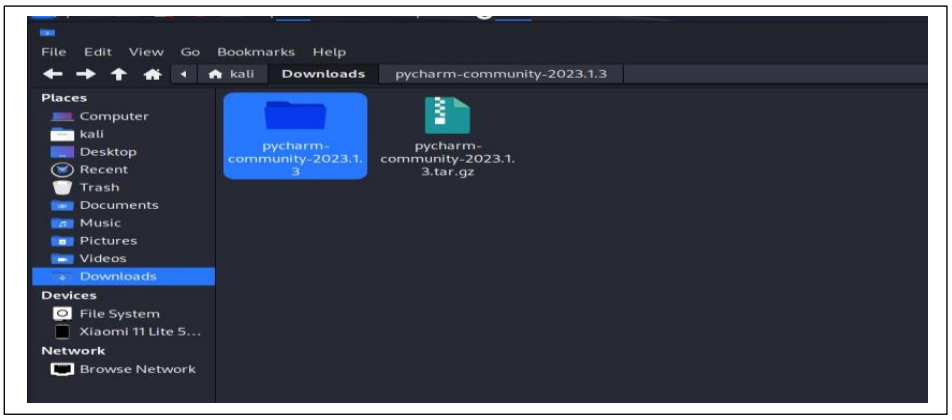


Figure III.2**:** fichier décompresser

e. On accède à ce fichier puis fait un clic droit et l'exécuter avec la termine. Apres on accède au fichier bin puis exécute la commande <<./pycham.sh>> pour lancer l'application pycham en acceptant tous les conditions).

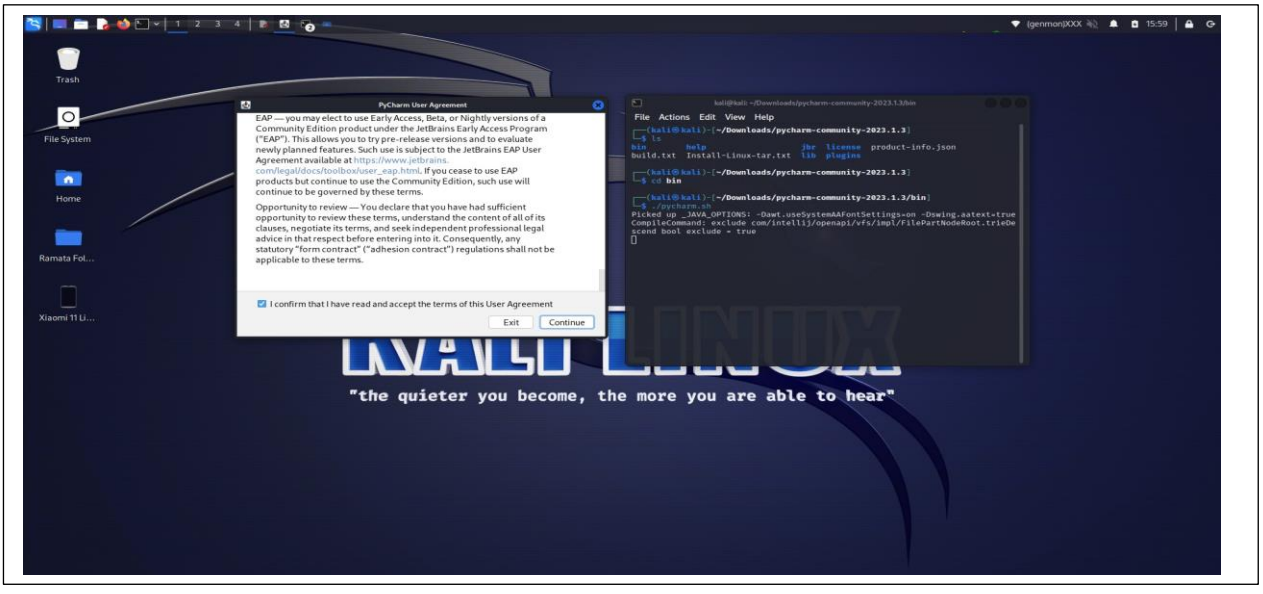


Figure III.3 **:** lancement de pycham

f. Création d'un nouveau Project pycharm en cliquant sur New project. Et nomme le project RSA-algorithme

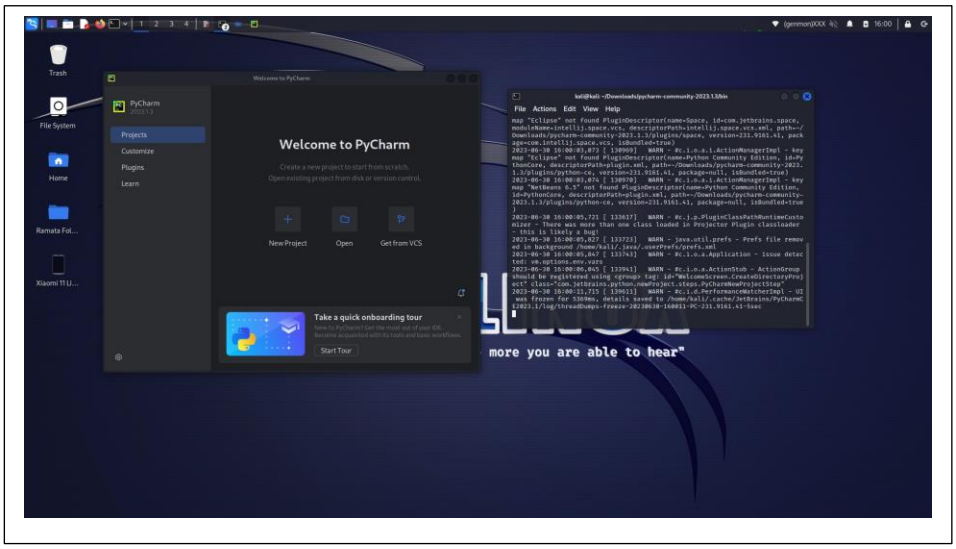


Figure III.4 : création d'un nouveau project

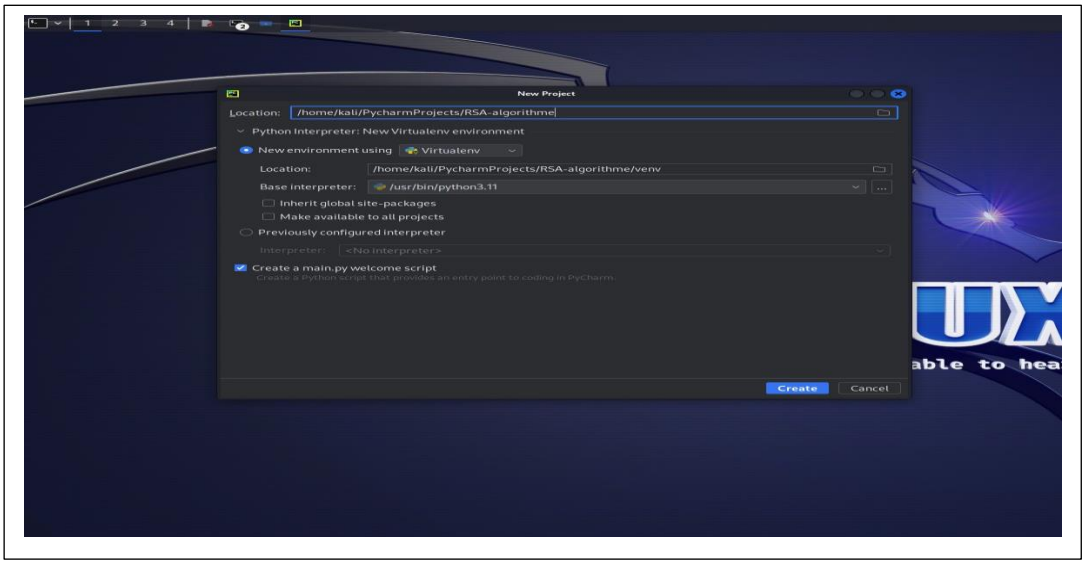


Figure III.5 : nom du Project en RSA-algorithme

g. On crée un fichier python qui contiendra toutes les méthodes de l'algorithme RSA (de la génération des clés au déchiffrement du fichier

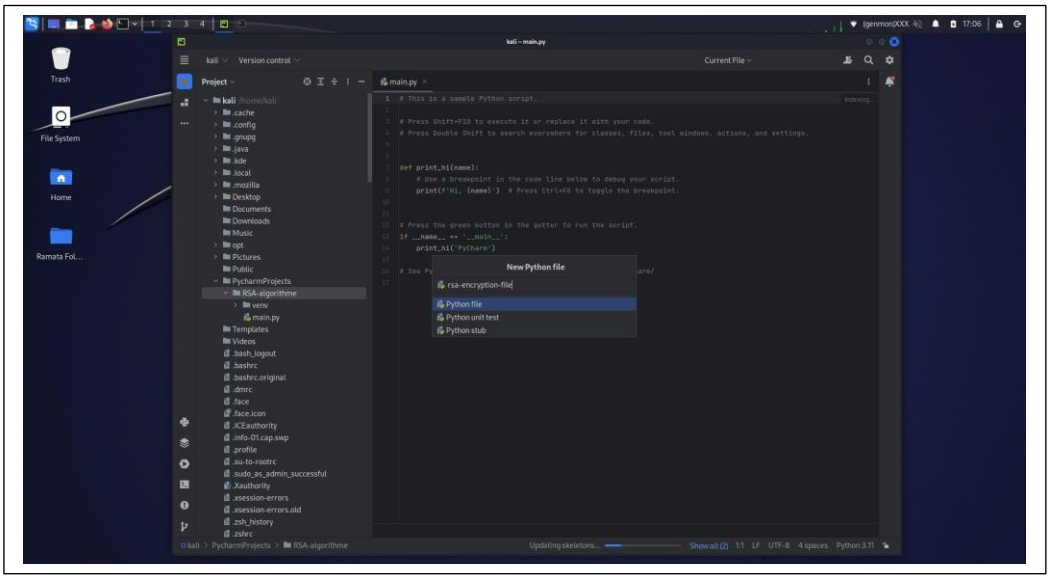


Figure III.6 : créer un fichier python rsa_encrytion_file.py

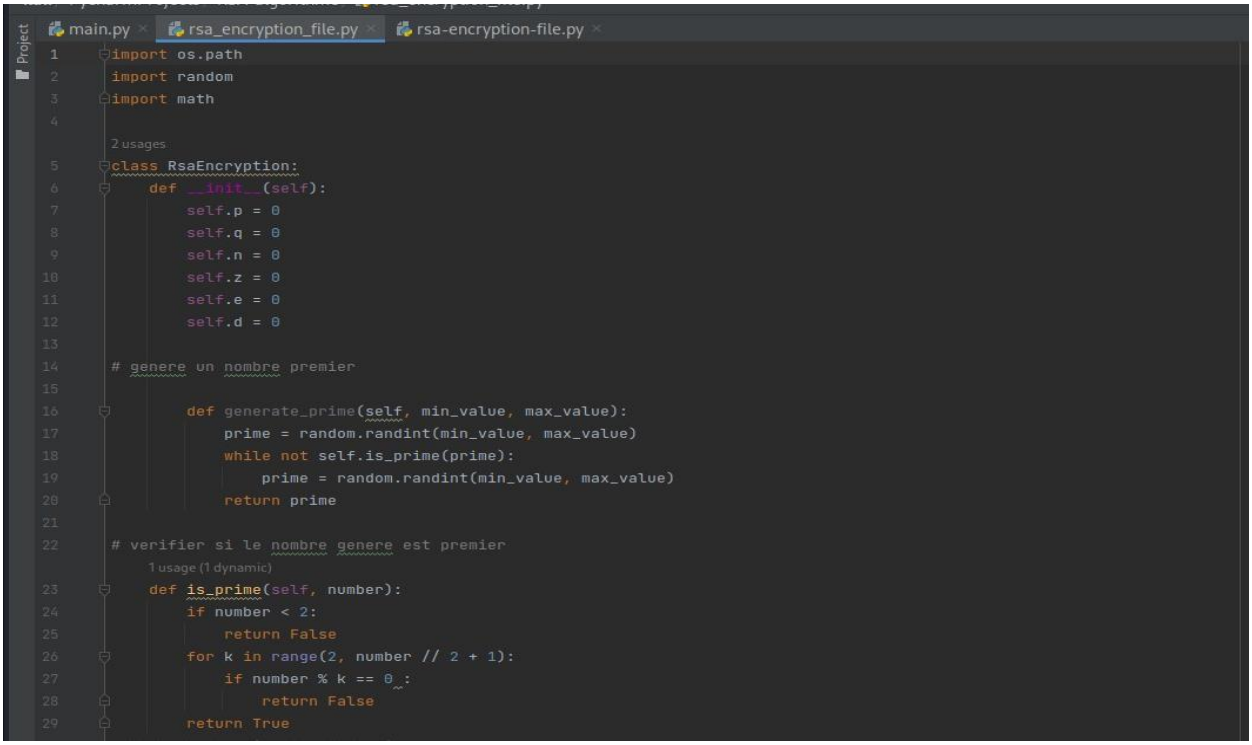
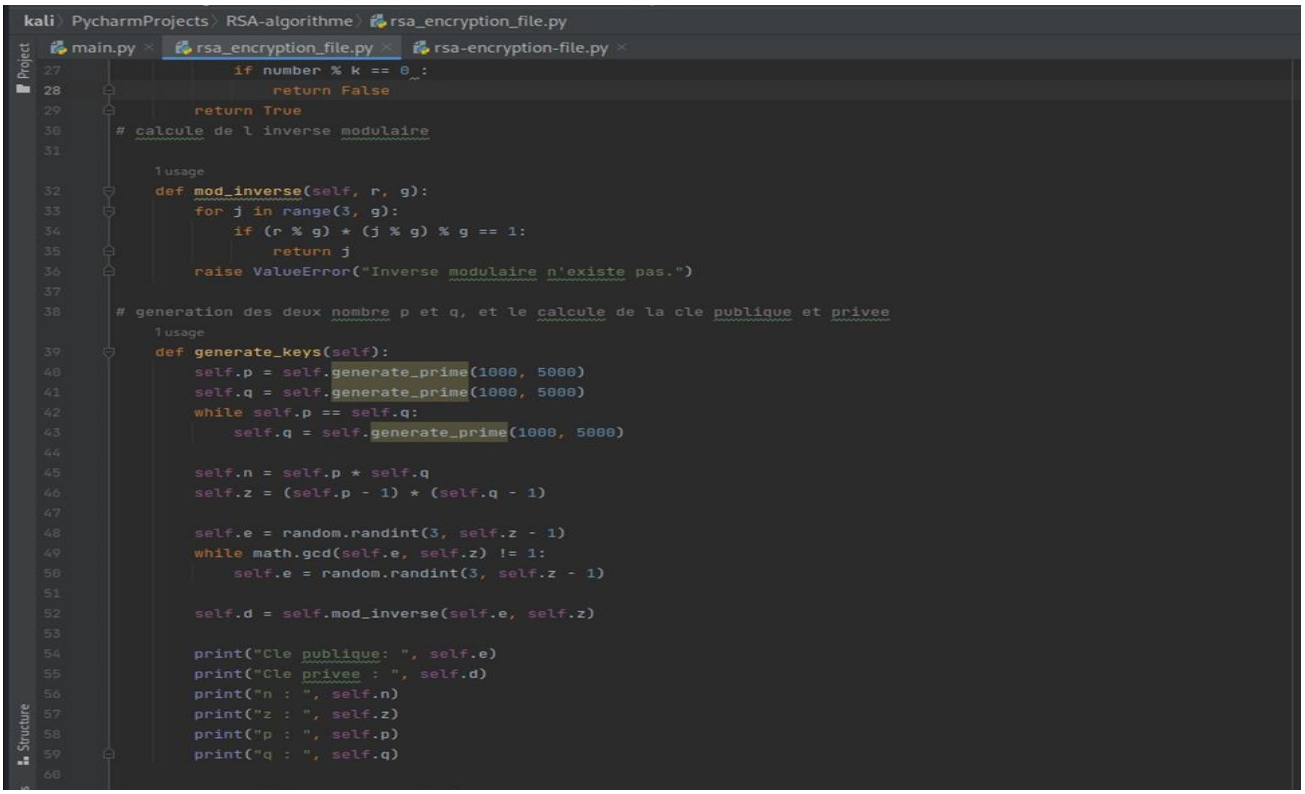
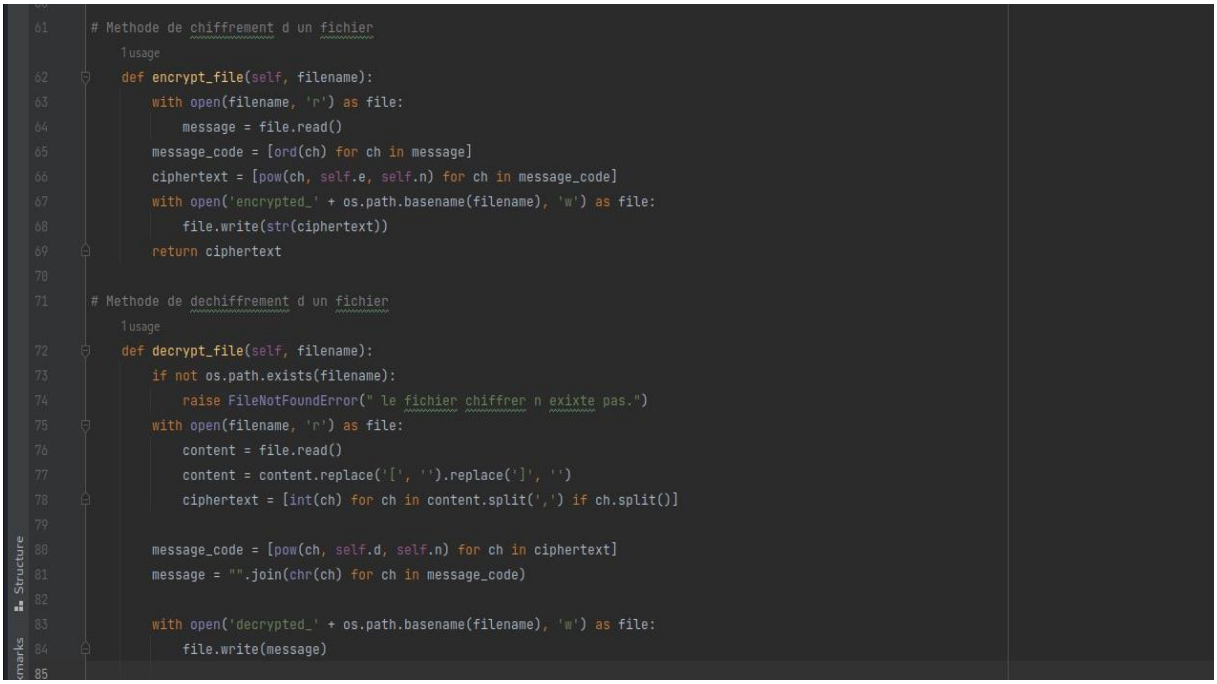


Figure III.7: les méthodes génération de clé et vérification de nombre premier



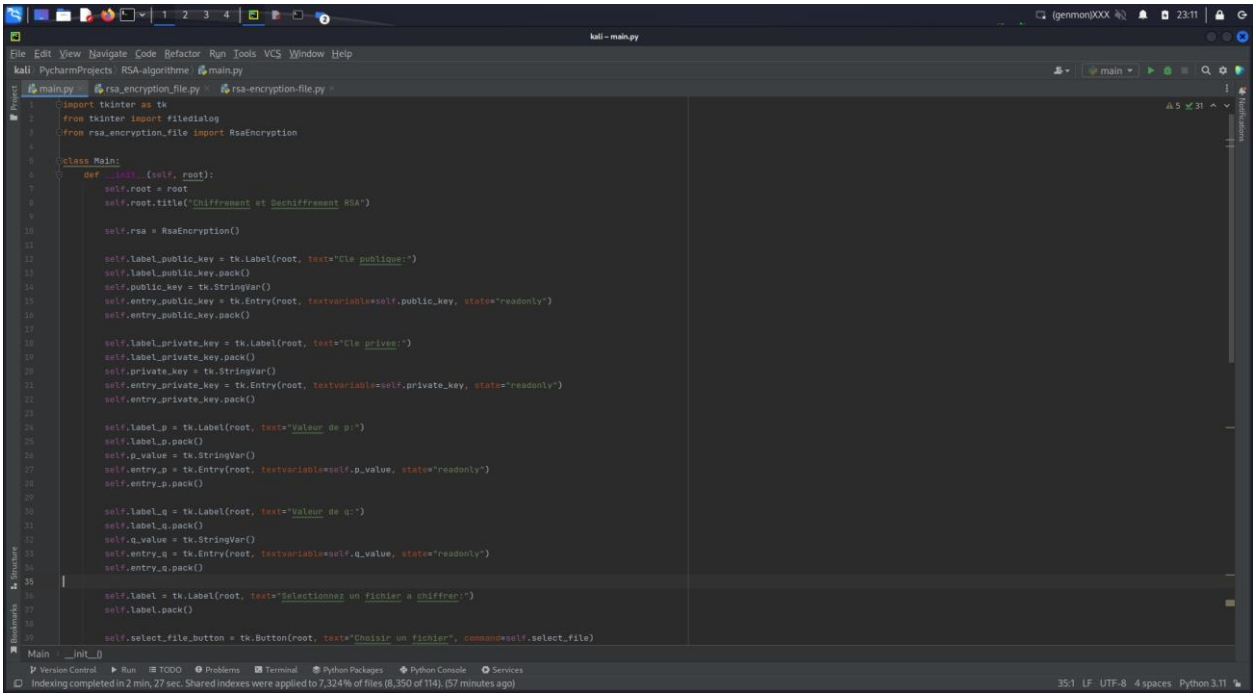

```
            if number % k == 0 :
                return False
        return True
# calcule de l inverse modulaire

    1 usage
    def mod_inverse(self, r, g):
        for j in range(3, g):
            if (r % g) * (j % g) % g == 1:
                return j
        raise ValueError("Inverse modulaire n'existe pas.")

# generation des deux nombre p et q, et le calcule de la cle publique et privee
    1 usage
    def generate_keys(self):
        self.p = self.generate_prime(1000, 5000)
        self.q = self.generate_prime(1000, 5000)
        while self.p == self.q:
            self.q = self.generate_prime(1000, 5000)

        self.n = self.p * self.q
        self.z = (self.p - 1) * (self.q - 1)

        self.e = random.randint(3, self.z - 1)
        while math.gcd(self.e, self.z) != 1:
            self.e = random.randint(3, self.z - 1)

        self.d = self.mod_inverse(self.e, self.z)

        print("Cle publique: ", self.e)
        print("Cle privee : ", self.d)
        print("n : ", self.n)
        print("z : ", self.z)
        print("p : ", self.p)
        print("q : ", self.q)
```

Figure III.8 : méthode du calcul de l'inverse modulaire et de générer les clés aléatoirement p et q entre (1000 et 5000)



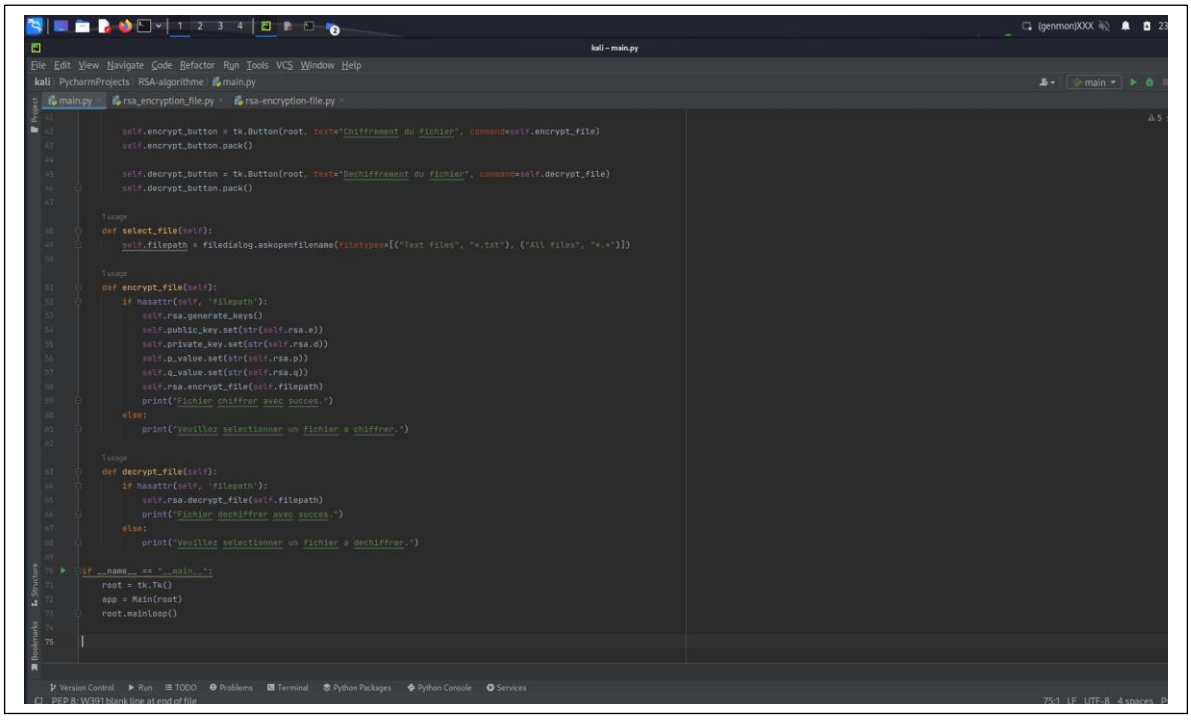
```
# Methode de chiffrement d un fichier
    1 usage
    def encrypt_file(self, filename):
        with open(filename, 'r') as file:
            message = file.read()
        message_code = [ord(ch) for ch in message]
        ciphertext = [pow(ch, self.e, self.n) for ch in message_code]
        with open('encrypted_' + os.path.basename(filename), 'w') as file:
            file.write(str(ciphertext))
        return ciphertext

# Methode de dechiffrement d un fichier
    1 usage
    def decrypt_file(self, filename):
        if not os.path.exists(filename):
            raise FileNotFoundError(" le fichier chiffrer n exixte pas.")
        with open(filename, 'r') as file:
            content = file.read()
            content = content.replace('[', '').replace(']', '')
            ciphertext = [int(ch) for ch in content.split(',') if ch.split()]

        message_code = [pow(ch, self.d, self.n) for ch in ciphertext]
        message = "".join(chr(ch) for ch in message_code)

        with open('decrypted_' + os.path.basename(filename), 'w') as file:
            file.write(message)
```

Figure III.8 **:** méthode chiffrement et déchiffrement du fichier



```python
import tkinter as tk
from tkinter import filedialog
from rsa_encryption_file import RsaEncryption

class Main:
    def __init__(self, root):
        self.root = root
        self.root.title("Chiffrement et Dechiffrement RSA")

        self.rsa = RsaEncryption()

        self.label_public_key = tk.Label(root, text="Cle publique:")
        self.label_public_key.pack()
        self.public_key = tk.StringVar()
        self.entry_public_key = tk.Entry(root, textvariable=self.public_key, state="readonly")
        self.entry_public_key.pack()

        self.label_private_key = tk.Label(root, text="Cle privee:")
        self.label_private_key.pack()
        self.private_key = tk.StringVar()
        self.entry_private_key = tk.Entry(root, textvariable=self.private_key, state="readonly")
        self.entry_private_key.pack()

        self.label_p = tk.Label(root, text="Valeur de p:")
        self.label_p.pack()
        self.p_value = tk.StringVar()
        self.entry_p = tk.Entry(root, textvariable=self.p_value, state="readonly")
        self.entry_p.pack()

        self.label_q = tk.Label(root, text="Valeur de q:")
        self.label_q.pack()
        self.q_value = tk.StringVar()
        self.entry_q = tk.Entry(root, textvariable=self.q_value, state="readonly")
        self.entry_q.pack()

        self.label = tk.Label(root, text="Selectionnez un fichier a chiffrer:")
        self.label.pack()

        self.select_file_button = tk.Button(root, text="Choisir un fichier", command=self.select_file)
```

Figure III.9 : la méthode main



```python
        self.encrypt_button = tk.Button(root, text="Chiffrement du fichier", command=self.encrypt_file)
        self.encrypt_button.pack()

        self.decrypt_button = tk.Button(root, text="Dechiffrement du fichier", command=self.decrypt_file)
        self.decrypt_button.pack()

    def select_file(self):
        self.filepath = filedialog.askopenfilename(filetypes=[("Text files", "*.txt"), ("All files", "*.*")])

    def encrypt_file(self):
        if hasattr(self, 'filepath'):
            self.rsa.generate_keys()
            self.public_key.set(str(self.rsa.e))
            self.private_key.set(str(self.rsa.d))
            self.p_value.set(str(self.rsa.p))
            self.q_value.set(str(self.rsa.q))
            self.rsa.encrypt_file(self.filepath)
            print("Fichier chiffrer avec succes.")
        else:
            print("Veuillez selectionner un fichier a chiffrer.")

    def decrypt_file(self):
        if hasattr(self, 'filepath'):
            self.rsa.decrypt_file(self.filepath)
            print("Fichier dechiffrer avec succes.")
        else:
            print("Veuillez selectionner un fichier a dechiffrer.")

if __name__ == "__main__":
    root = tk.Tk()
    app = Main(root)
    root.mainloop()
```

Figure III.10 **:** suite de la méthode main

h. On passe maintenant l'exécution de la classe Main pour effectuer un test.

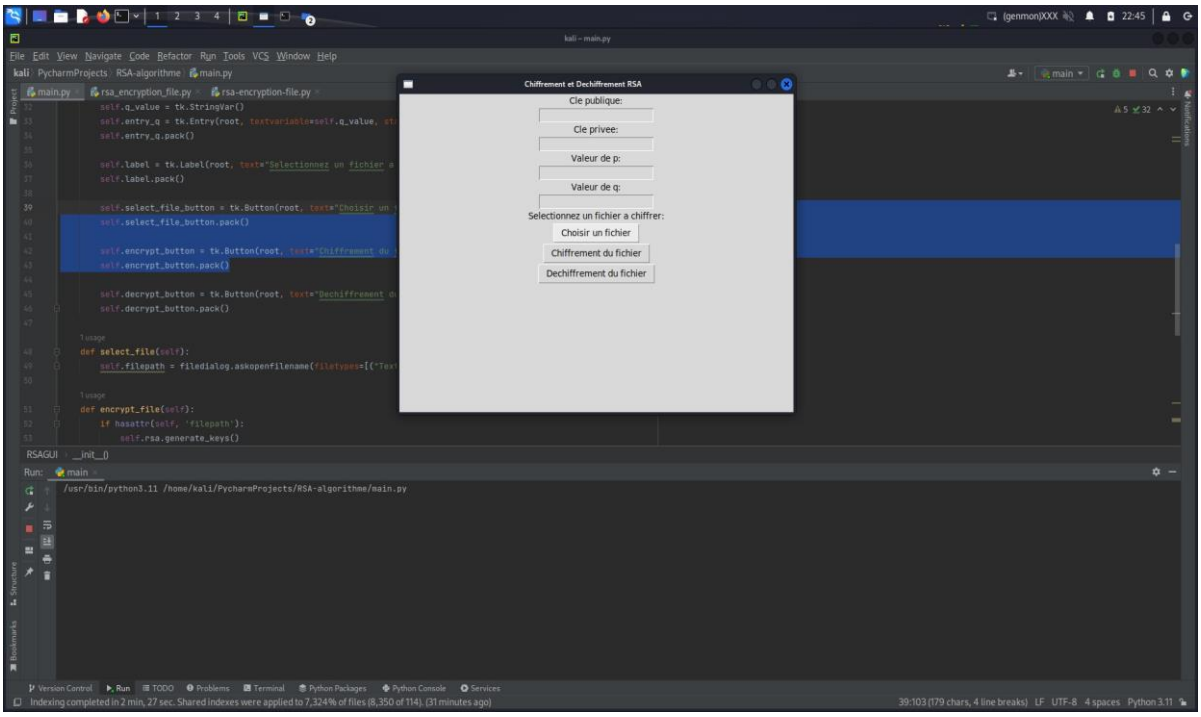


Figure III.11 : interface utilisateur

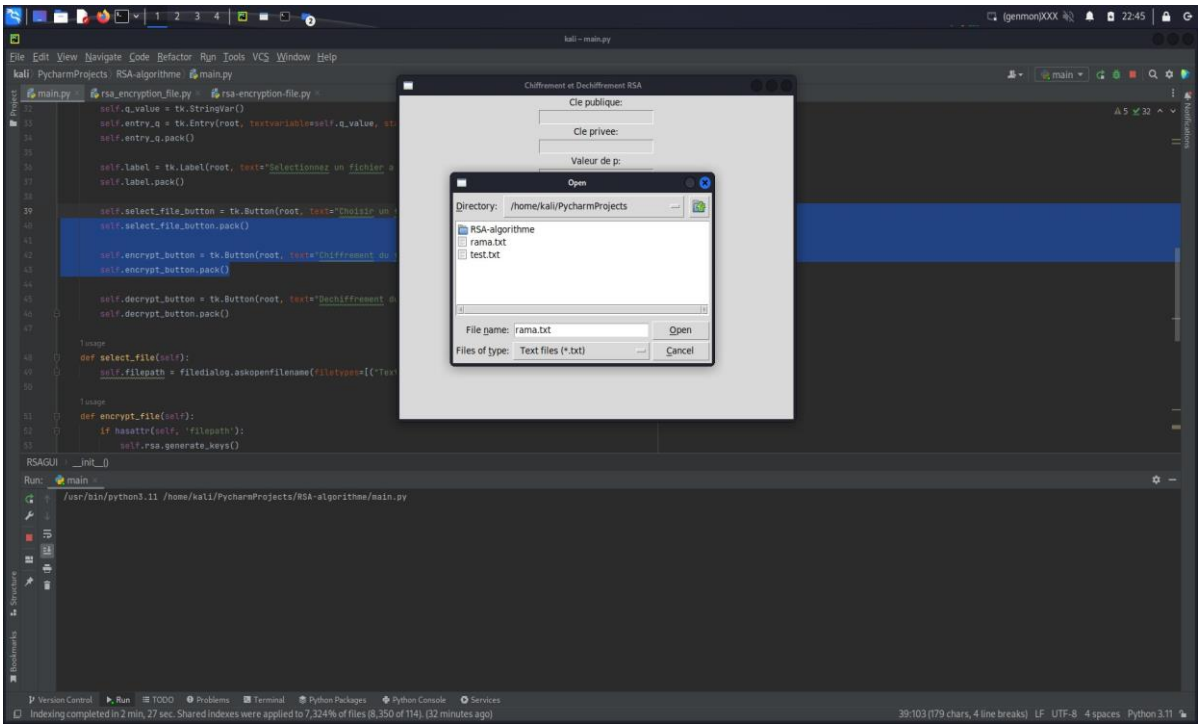


Figure III.12 : choisir un fichier.txt existant pour le chiffrer

Apres choit du fichier et l'appuis sur le bouton chiffrement la valeur des clés sont affichées et un fichier encrypt.txt est créer contenant le message chiffré

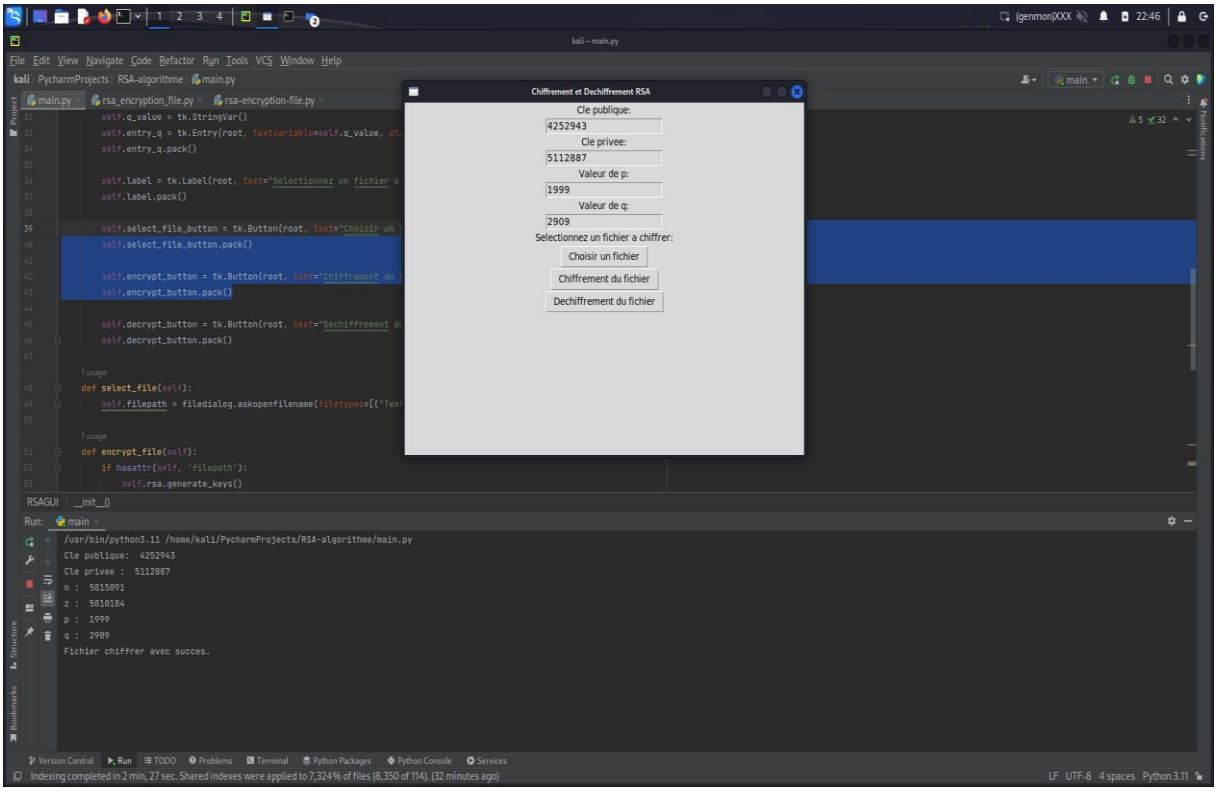


Figure III.13 **:** fichier chiffre

Pour le déchiffrement on choisit le fichier encrypt.txt

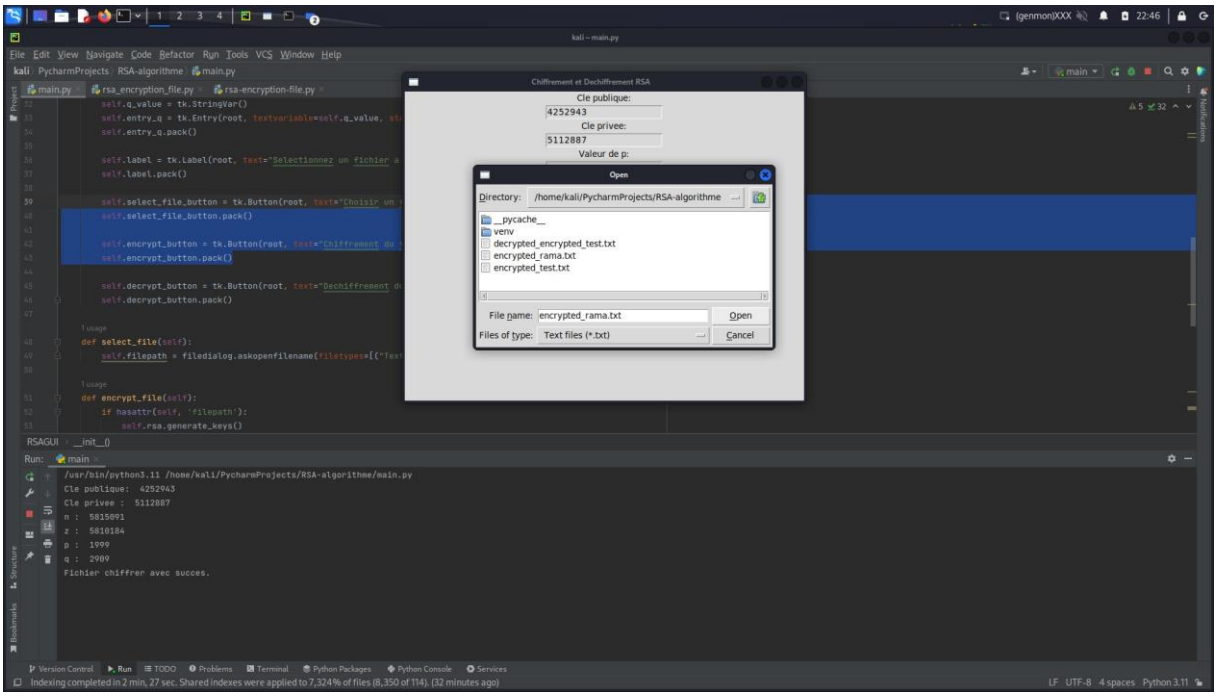


Figure III.14 : fichier chiffrer pour le déchiffrement

Apres avoir choisi le fichier chiffré pour le déchiffrement un message est affiche sur la console si le déchiffrement a réussi (fichier déchiffre avec succès), accompagner de la création du fichier decrypt.txt.

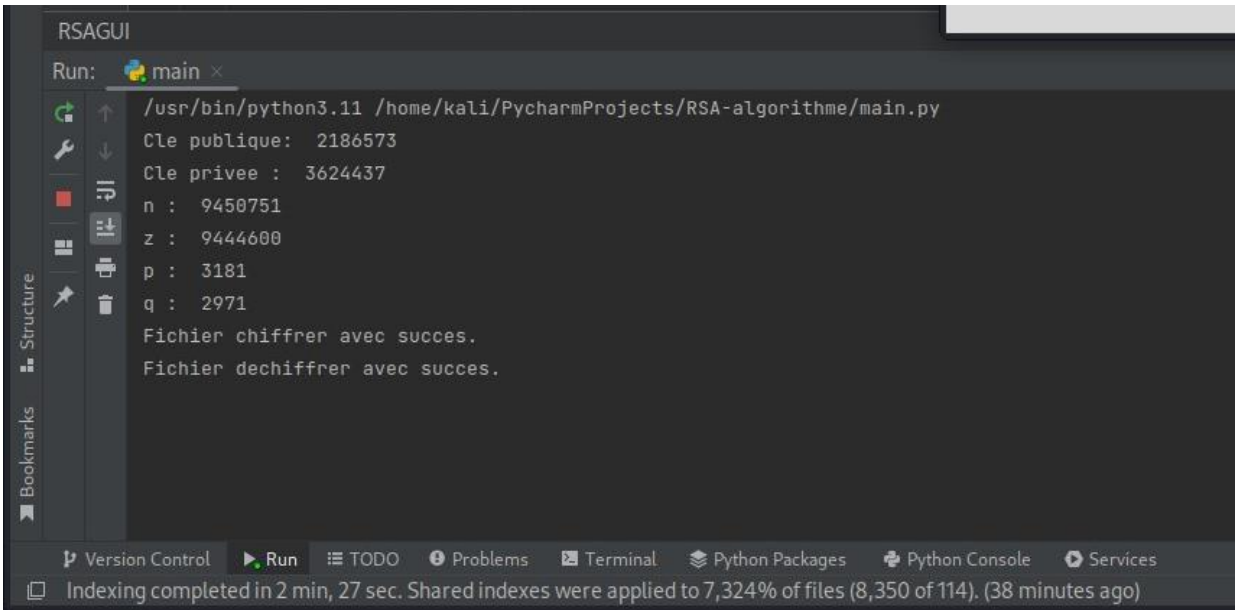


Figure III.15 : fichier déchiffre

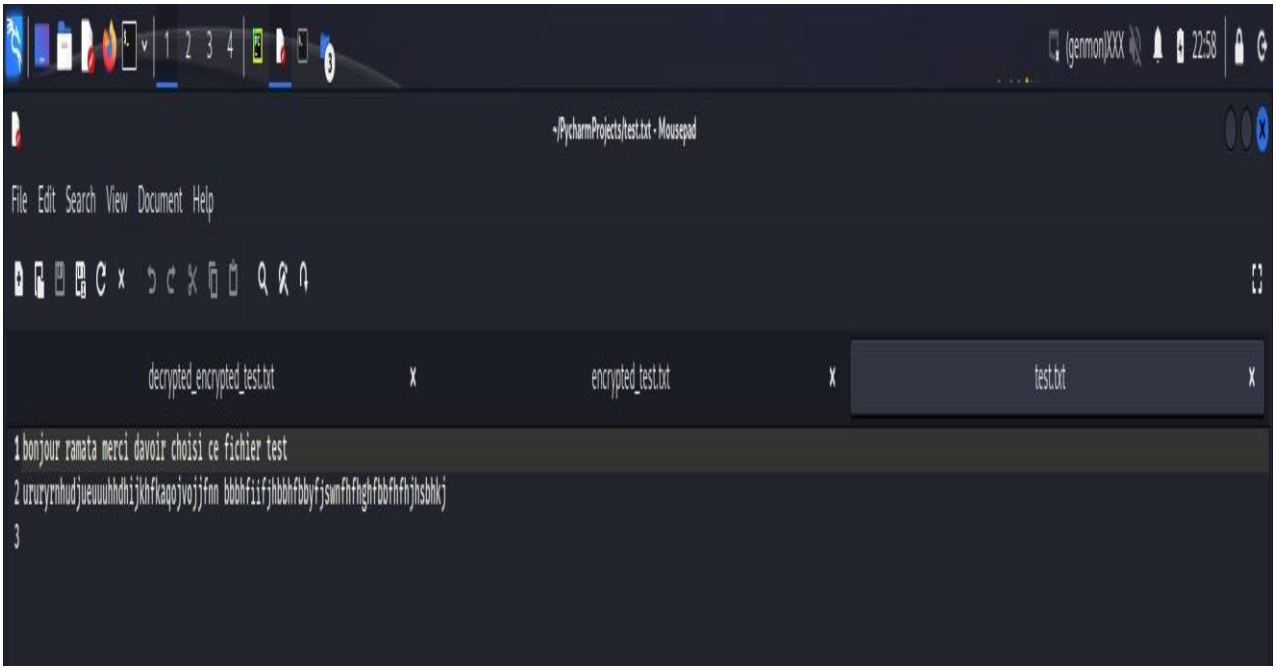


Figure III.16 : contenue du fichier test a chiffré

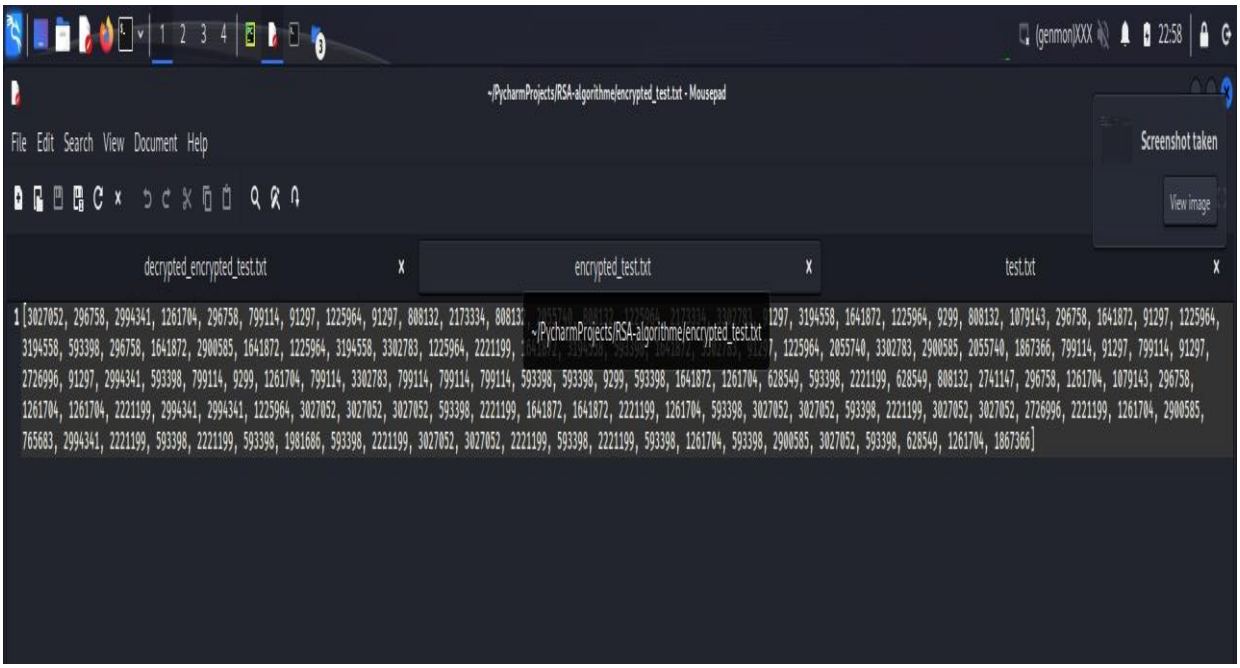


Figure III.17 : contenue du fichier test chiffré (encryption.txt)

Figure III.18 : conte nue du fichier test déchiffré (decryption.txt)

3.2 Craquage d'un réseau sans fil

3.2.1 Conception :

L'objectif est de tester la sécurité d'un réseau sans fil utilisant le protocole TKIP pour sécuriser ses communications sans fil, et voir le temps qu'il faut à un hacker de trouver la clé de celui-ci en fonction de sa taille allant de 8 à 16 chiffres.

3.2.2 TEST :

Test sur les nombres de caractère après avoir créé un fichier texte (password.txt) contenant les combinaisons des nombres de 0 à 9 de 8 à 16 caractères.

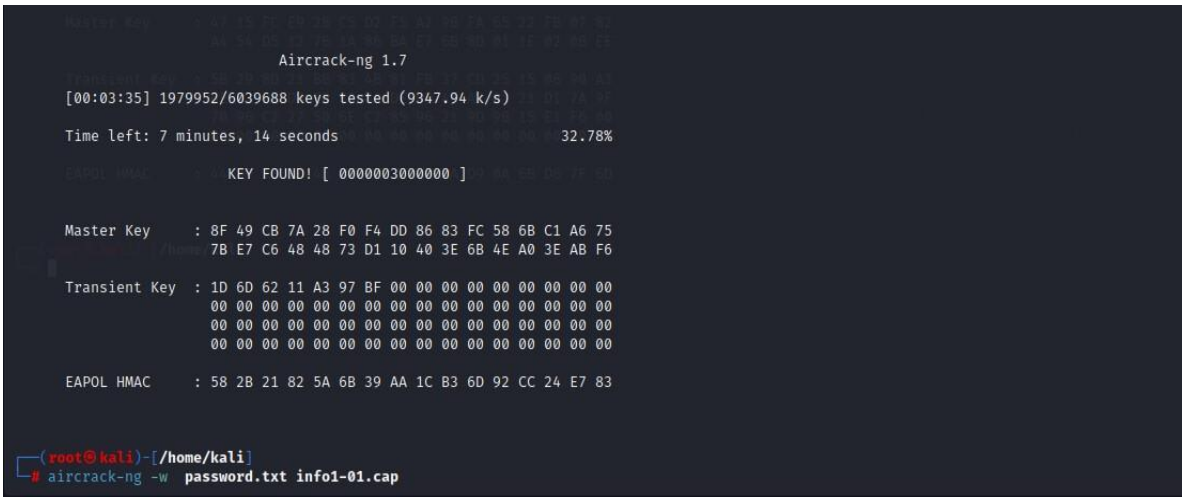


Figure III.19 : la clé comporte 8 chiffres

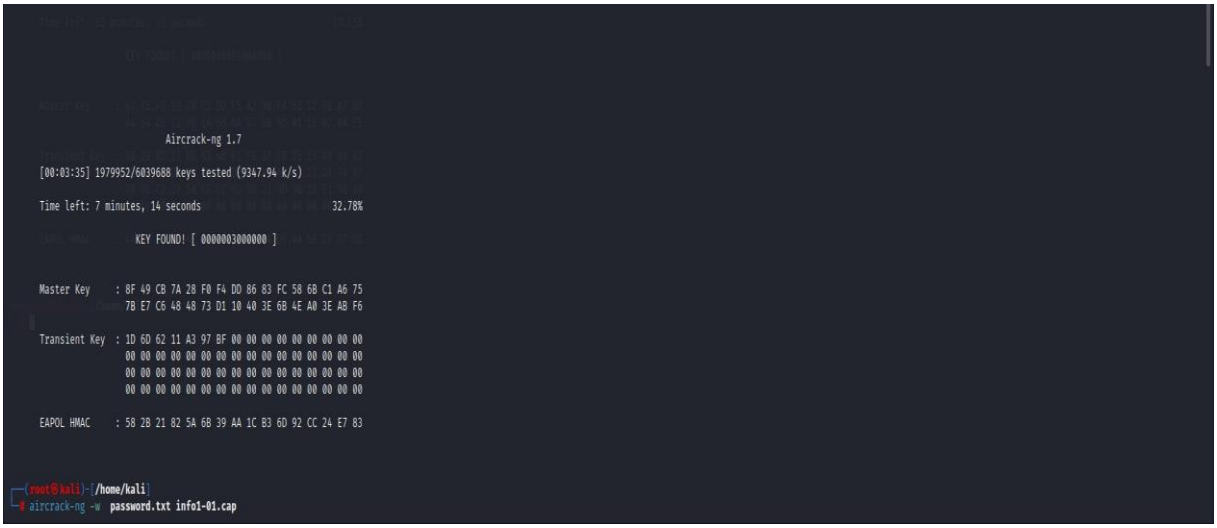


Figure III.20 : la clé comporte 13 chiffres

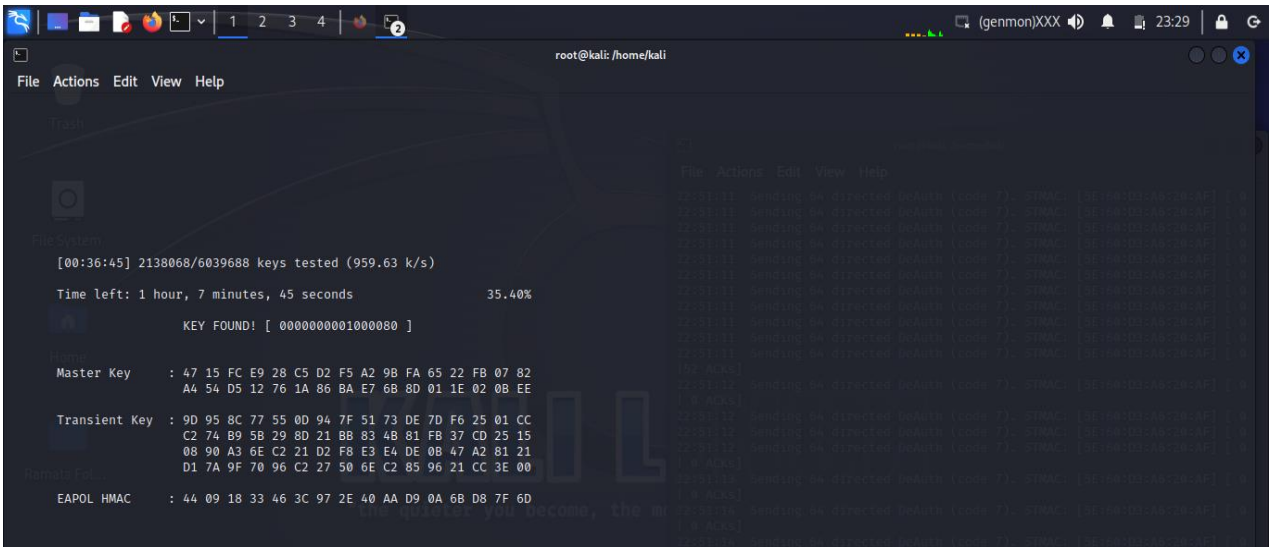


Figure III.21 : la clé comporte 16 chiffres

3.2.3 Wireshark :

C'est un outil de capture et d'analyse de paquets réseau. Son rôle est d'analyser en détail le contenu des paquets capturés dans un fichier PCAP. Il décode les protocoles, extrait les données, fournit des fonctionnalités de filtrage et de recherche, et présente des statistiques pour faciliter l'analyse du trafic réseau et la résolution des problèmes liés aux communications.

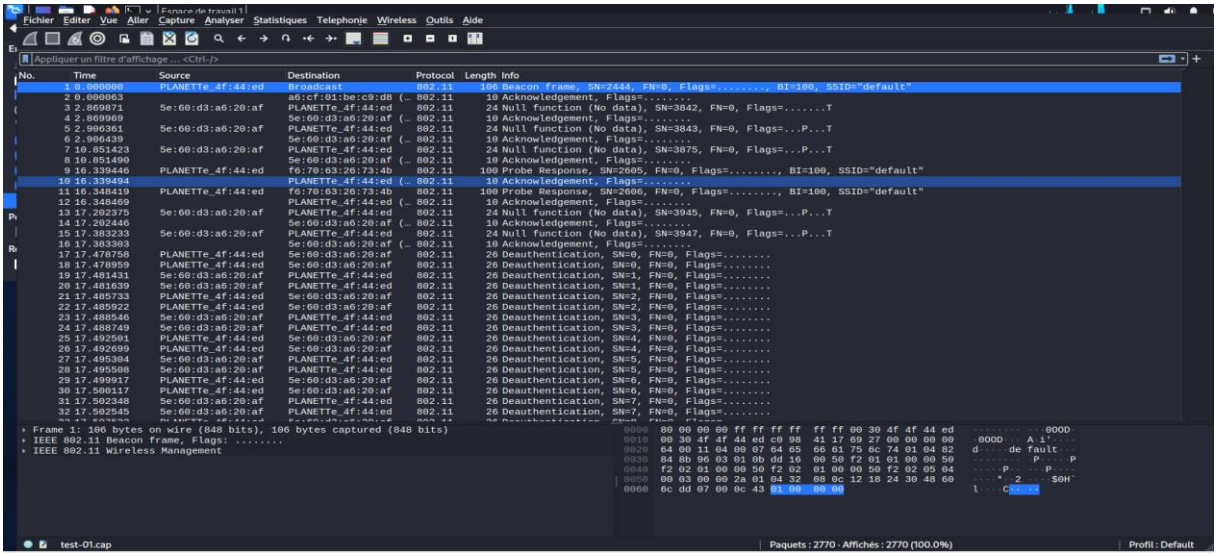


Figure III.23 : contenue du fichier.cap de l'exemple 8 caractères

CHAPITRE4

RÉSULTAT ET DISCUSSION

1. Temps de déchiffrement RSA

Apres la génération des clés l'application va afficher sur la console le temps qui faut à un fichier test.txt dont le contenu était (bonjour ramata merci davoir choisi ce fichier test) pour être déchiffrer en fonction des changements des valeurs de p et q.

P	Q	N	E	D	Temps (seconde)
73	83	6059	2521		0,00008
101	107	10807	7309	2989	0,000097
367	691	253597	226901	118661	0,000163
571	821	468791	397039	242359	0,000176
3343	5741	19192163	6283481	1868441	0,000197
19427	21841	424305107	158164861	304653301	0,000271
46237	35509	1641829633	903333599	1007893583	0,000364
59069	76403	4513048807	3953549295	19920125543	0,00067

Tableau IV.1 : Temps nécessaire pour le déchiffrement RSA

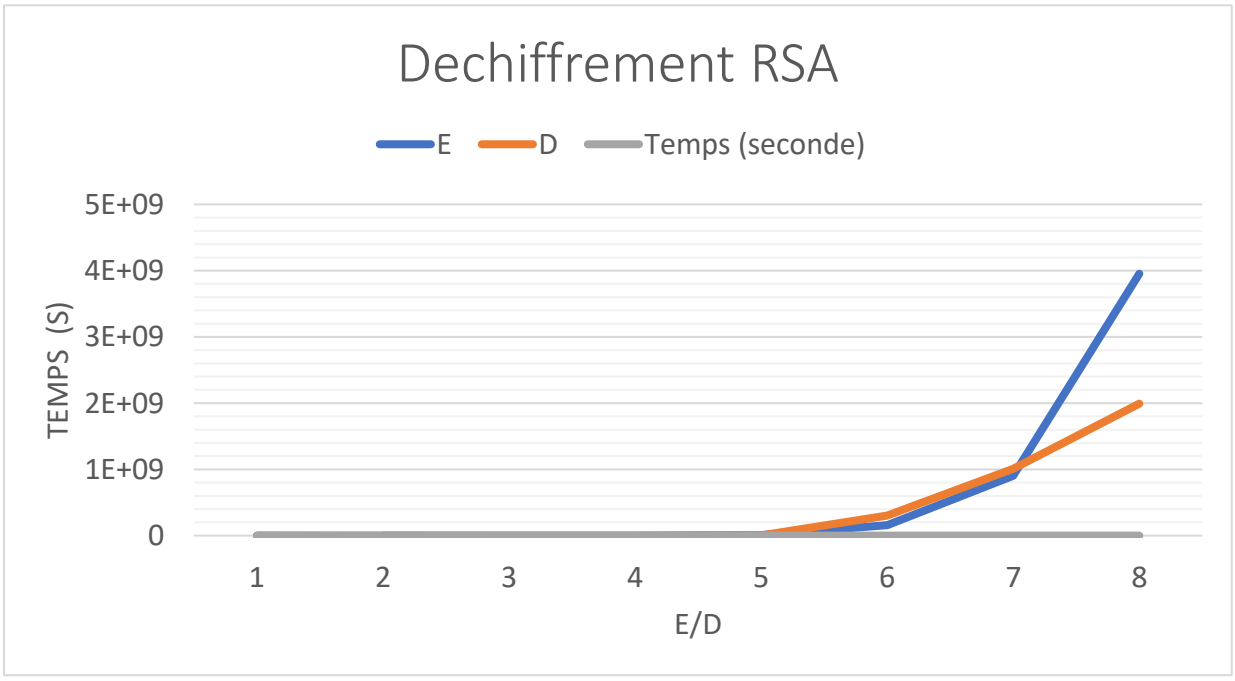


Figure IV.1 : Graphe des valeurs de E/D en fonction du temps

2. Temps de chiffrement TKIP

A la fin du lancement de aircrack-ng le temps de chiffrement est affiché avec le mot de passe trouver.

Longueur de mots de passe	Temps en s
8	1
9	18
10	20
11	28
12	32
13	215
14	1800
15	1980
16	2205

Tableau IV.2 **:** Temps nécessaire pour le craquage TKIP

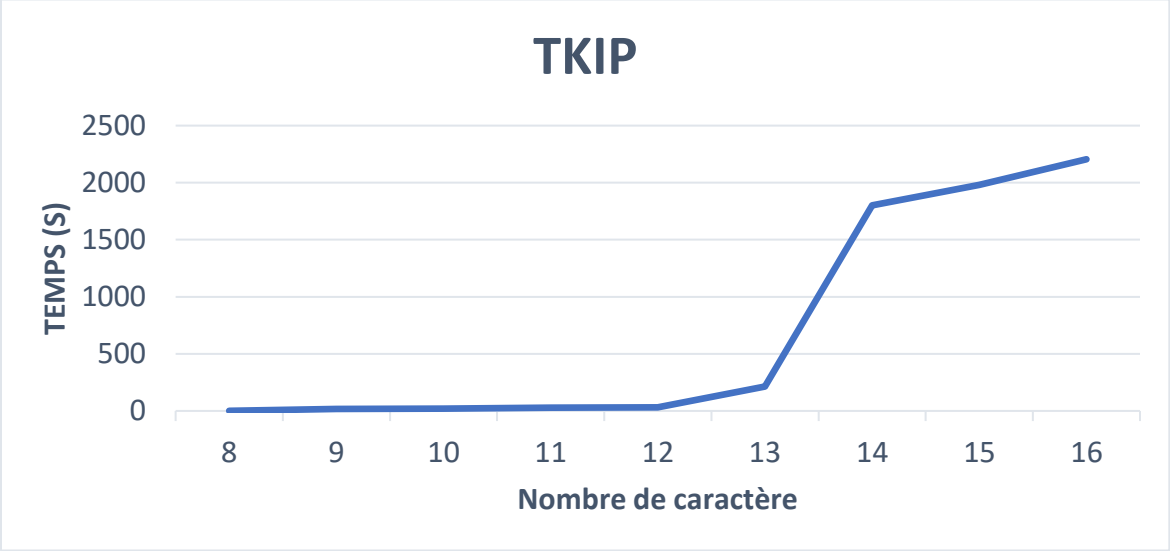


Figure IV.2 **:** Graphe du temps calculé pour le chiffrement **TKIP**

3. Discussion

D'après les graphes de temps pour les deux scénarios, on remarque que dans les deux cas plus la taille de la clé augmente, plus le temps pour leur réaliser est important.

4. Conclusion

Dans ce chapitre nous avons présenté les résultats des différents tests effectués au niveau de RSA et TKIP, et celui-ci montre que leur sécurité est liée à la taille de la clé, plus la clé est grande, plus le risque diminue.

Conclusion générale

Il est clair que ces deux protocoles de chiffrement présentent des caractéristiques distinctes en termes de sécurité et de vulnérabilités.

D'une part, le protocole RSA, basé sur la cryptographie asymétrique, est largement utilisé pour le chiffrement des données sensibles. Son principe repose sur l'utilisation de deux nombres premiers très grands, ce qui rend la factorisation de la clé publique extrêmement difficile, voire impossible avec les ressources informatiques actuelles. Malgré quelques vulnérabilités liées à des choix erronés de paramètres, RSA reste un algorithme robuste et fiable pour la sécurisation des communications.

D'autre part, le protocole TKIP, utilisé dans les réseaux sans fil basés sur la norme IEEE 802.11, présente des vulnérabilités plus importantes. Les attaques sur TKIP, telles que l'attaque de réinitialisation de paquets, l'attaque de fragmentation et l'attaque par dictionnaire, ont montré qu'il était possible de compromettre la sécurité du protocole et de récupérer des données sensibles.

Cependant, il convient de noter que la comparaison des difficultés de craquage entre RSA et TKIP doit tenir compte de plusieurs facteurs. La taille des clés utilisées dans chaque protocole, les avancées technologiques dans le domaine de la cryptanalyse, ainsi que les ressources et les capacités de l'attaquant jouent tous un rôle crucial dans la détermination de la difficulté de craquage.

Dans l'ensemble, RSA offre une meilleure sécurité que TKIP en raison de sa nature basée sur la factorisation des nombres premiers. Cependant, il est essentiel de souligner que la sécurité de tout protocole dépend de la mise en œuvre correcte, de la gestion des clés et des mesures de sécurité supplémentaires mises en place. Donc la comparaison des difficultés de craquage entre RSA et TKIP met en évidence la nécessité de choisir le protocole de chiffrement approprié en fonction des besoins spécifiques de sécurité, du contexte d'utilisation et des ressources disponibles. La cryptographie continue d'évoluer, et il est essentiel de rester à jour avec les dernières avancées et les meilleures pratiques pour garantir une sécurité optimale des communications.

BIBLIOGRAPHIE

[1] Cherif.Zahar. Cours de Cryptographie, 3ème année de licence, USDB1, 2021

[2] L.Z.Ismail & E.Youcef, ‘’SIMULATION D’UNE ATTAQUE SUR LE CRYPTOSYSTEME RSA’’, mémoire de master, UNIVERSITE ABDELHAMID IBN BADIS MOSTAGANEM, 2013

[3] K.Yamina & Z.Cylia, ‘’ Simulation de quelques attaques sur le crypto-système RSA’’, mémoire de master, Université Mouloud Mammeri De Tizi-Ouzou, 2020

[4] G.Yasmine, ‘’ cour de sécurité ‘’, master1, USDB1, 2021

[5] S.Zakaria‘’ cour de sécurité des systèmes informatiques ‘’, master2, USDB1, 2023

[6] Jean-françois pillou, Cryptographie, Mai 2015
https://www.commentcamarche.net/contents/203-cryptographie

[7] Bastien L, Sécurité, 4 avril 2019
https://www.lebigdata.fr/chiffrement-des-donnees-tout- savoir

[8] Culture Informatique, Comment ça marche le cryptage ? 18 mars 2016
https://www.culture-informatique.net/comment-ca-marche-cryptage

[9] Preuve de sécurité https://fr.wikipedia.org/wiki/Preuve_de_sérurité

[10] Explication, Décryptage https://chiffrer.info

[11] Amini.H & Guermah.N, ‘’ETUDE SUR LES MECANISMES DE SECURITE D’UN RESEAU WIFI’’, mémoire de master, UNIVERSITE MOULOUD MAMMERI TIZI-OUZOU ,2013

[12] https://cisco.goffinet.org/ccna/wlan/protocoles-securite-sans-fil-wpa-wpa2-wpa3

[13] What is TKIP https://fr.theastrologypage.com/temporal-key-integrity-protocol#menu-3

[14] ‘’Sécurité Wi-Fi–WEP,WPA et WPA2’’ https://repo.zenk-security.com/Protocoles_reseaux_securisation/Securite%20Wi-Fi%20-%20WEP,%20WPA%20et%20WPA2.pdf

[15] A.C.Salah & G.Zinaba, ‘’ Attaques aux protocoles Wi-FI ‘’ , mémoire de master, Université de Larbi Tébessi –Tébessa, 2016