Table des matières

Liste des figures

Liste des tableaux

Listes des abréviations, sigles et acronymes

ACL: Access Control List.

API: Application Programming Interface.

AMS : Agent Management System.

BS : Bande Spectrale.

CA : Autorité de certification (Certificate Authority).

CCS : Saturation des canaux de contrôle (Control Channel Saturation DoS).

CPU : Coalition Primary User

CSMA : Détection de porteuse par accès multiple (Carrier Sensing Multiple Access).

CSU : Coalition Secondary User

DDT : Test de différence de distance (Distance difference Test).

DF : Directory Facilitator.

DoS : Déni de service (Denial of Service).

DRT : Test du rapport de distance (Distance Report Test).

EAP : Extensible Authentication Protocol.

EMS : Signatures électromagnétique (Electromagnetic Signatures).

FCC : Commission fédérale des communications (Fédéral Communications Commission).

FIPA: Foundation for Intelligent Physical Agents

FR : Fréquence Radio.

GPS : Système de positionnement global (Global Positioning System).

GSM : Groupe spécial mobile (Global System for Mobile Communications).

HSPA : Accès par paquets haute vitesse (High Speed Packet Access).

IA : Information d'authentification.

IDS : Système de détection d'intrusion (Intrusion Detection System).

ISMS : Système de gestion de la sécurité de l'information (Information Security Management System).

JADE : Java Agent Development Framework.

JAVA : Langage de programmation informatique orienté objet.

KTH : Institut royal de technologie **(**Kungliga Tekniska högskolan).

LLC : Vérification de la cohérence de l'emplacement (Location Consistency Checks).

LocDef : Défense basée sur la localisation (Localization Based Defense).

LTE : Evolution à long terme (Long Term Evolution).

MAC : Couche de contrôle d'accès au support (Media Access Control).

MSU : Utilisateur secondaire malveillant (Malicious Secondary User)

OSI : Modèle de communication entre ordinateurs **(O**pen **S**ystem **I**nterconnection).

PDR : Ratio de livraison des paquets (Packet Delivery Ratio).

PU : Utilisateur primaire (Primary User).

PUE : Emulation de l'utilisateur primaire (Primary User Emulation).

RC : Radio Cognitive.

RFF : Imagerie numérique par radiofréquence (Radio Frequency Fingerprinting).

RSS : Résistance du signal reçu (Received Signal Strength).

RL : Radio Logicielle.

RRC : Réseau de la Radio Cognitive.

SCN : Négociation de canal égoïste (Selfish Channel Negociation)

SDR : Radio logicielle Restreinte (Software Defind Radio).

SEAD : Protocole de routage ad hoc à vecteur de distance (Secure Efficient Ad hoc Distance Vector).

SDF : Plusieurs fusion de données (Several Data Fusion).

SMA : Système Multi-agents

SS : Force du signal (Signal Strength).

SSDF : Falsification des données de détection du spectre (Spectrum Sensing Data Falsification).

SU : Utilisateur secondaire (Secondary User).

TCP : Protocole de contrôle de transmissions (Transmission Control Protocol).

WLAN: Réseau local sans fil (Wireless Local Area Network).

WSN : Réseau de capteurs sans fil (Wireless Sensor Network).

WSPRT : Test pondéré de rapport de probabilité séquentiel (Weighted Sequential Probability Ratio Test).

WSRT : Test pondéré de rapport séquentiel (Weighted Sequential Ratio Test).

Introduction générale

Les services sans fil augmentent de plus en plus ce qui conduit à une demande importante des communications sans fil ce qui rend le spectre encombré.

Les chercheurs développent de nouvelles technologies pour résoudre le problème des espaces libres du spectre. Cette nouvelle technologie est la radio cognitive (RC). La radio cognitive utilise le spectre d'une manière opportuniste qui permet la détection des canaux de communications d'une façon intelligente.

Le principe de la radio cognitive est basé sur les utilisateurs primaires qui ont des bandes de fréquences attribuées et les utilisateurs secondaires qui ne possèdent pas de licence. Selon la technologie de la radio cognitive, les utilisateurs secondaires utilisent les espaces libres des bandes de fréquences des utilisateurs primaires.

Un dispositif radio cognitif utilise des processeurs informatiques à usage général qui exécutent un logiciel d'application radio pour effectuer le traitement du signal. L'utilisation de ce logiciel permet à l'appareil de sentir et de comprendre son environnement et de changer activement son mode de fonctionnement en fonction de ses observations. Malheureusement, cette solution implique de nouveaux problèmes de sécurité car elle est ciblée par des différentes attaques d'une façon simple et rapide. Donc, La sécurité dans un réseau radio cognitif est très importante pour l'utilisation du spectre d'une façon fiable.

Notre contribution dans le cadre de ce mémoire consiste à instaurer un algorithme qui sécurise la communication pour l'accès dynamique au spectre dans un réseau radio cognitif c'est-à-dire la communication entre les PU et les SU doit être sécurisée, nous devons assurer que les SU communiquent et demandent l'allocation des bandes de fréquences à des PU légitimes pour que les données soient fiables.

Tout d'abord, nous allons utiliser un algorithme d'authentification Blowfish et un autre de la décision multicritère TOPSIS. Puis, nous allons réaliser deux méthodes pour sécuriser le réseau radio cognitif. Par la suite, nous allons comparer les deux algorithmes afin de mesurer la qualité de service et surtout la fiabilité.

Dans le premier chapitre, nous allons définir la radio logicielle, le concept de la radio cognitive et les différentes fonctions utilisées par cette technologie.

Dans le deuxième chapitre, nous allons présenter les généralités sur la sécurité dans les réseaux sans fil et nous nous concentrerons sur les attaques contre le réseau radio cognitif et les solutions proposées.

Dans le troisième chapitre, nous allons décrire et présenter notre algorithme et les différentes approches utilisées pour réaliser notre travail. A la fin, nous allons montrer une étude comparative sur les résultats obtenus.

CHAPITRE I

Radio Cognitive

I. CHAPITRE I : Radio cognitive

I.1. Introduction

En 1907, l'inventeur et physicien italien Guglielmo Marconi réalise le premier système télégraphe sans fil qui traverse l'océan atlantique. L'utilisation des systèmes de communication sans fil et de l'internet mobile augmente d'une manière incroyable la raison pour laquelle les chercheurs essayent toujours à améliorer les performances de ces réseaux.

En 1998, Joseph Mitola III présente la notion de la Radio Cognitive RC à l'institut royal de technologie KTH[1]et en 1999 l'article a été publié. Mitola a réussi d'assembler entre sa recherche de la Radio Logicielle et l'Intelligence Artificielle pour introduire la technique de la radio cognitive.

En mai 2003, la Commission fédérale des communications (FCC) des États-Unis d'Amérique a convoqué des chercheurs pour créer un atelier, dans laquelle l'impact de la RC sur l'utilisation spectrale est examiné et les problèmes liés à la réglementation de la RC sont étudiés. Au cours de cet atelier, Laren Van Wazer, conseiller spécial chez le bureau d'ingénierie et technologie de la FCC, a dit que la FCC désire « améliorer l'accès aux bandes spectrale (BS) à travers une meilleure utilisation du temps, de la fréquence, de la puissance, de la bande passante et l'espace »[1] [2] [3]

I.2. Définition

I.2.1. Radio logicielle : Software Radio

Selon Mitola, les éléments matériels de la radio tels que la fréquence porteuse, la largeur de bande de signal, la modulation et l'accès aux réseaux sont effectués sous forme de logiciel.

[1] KTH est une grande école d'ingénieurs de Stockholm en Suède. Elle a été fondée en 1827 et est la plus grande université de technologie scandinave.

Actuellement, la radio insert aussi des techniques de la cryptographie telles que le codage et correction d'erreur. L'idée de la radio logicielle permet d'emmètre les utilisateurs et les créateurs plus autonomes aux règles.

Le principe de la Radio Logicielle doit être approprié à un ensemble de critères : la nécessité d'un service pour un client exclusif dans un entourage offert et pendant une durée fixe.

La Radio Logicielle introduit une couche intermédiaire dite Radio Logicielle Restreinte SDR qui permet la communication radio afin d'avoir la puissance, le calcul, la consommation électrique et le coût … etc., de la bande de fréquence.[4]

I.2.2. Radio Cognitive

La radio cognitive est un aspect de liaison sans fil qui permet la découverte des canaux de communications pendant l'utilisation et qui ne sont pas utilisés, et le déplacement dans les canaux inemployés, cette approche permet la détection et la reconnaissance de ces limites d'application pour adapter les grandeurs d'activité radio d'une manière indépendante et de savoir les bilans de ces œuvres et son entourage de fabrication.

La Radio Cognitive c'est le fait de mieux utiliser le spectre (vue qu'il y a certaine bande de fréquence surchargées et d'autre non utilisées) et cela implique d'augmenter le débit et assumer la fiabilité de la couche physique.

La Radio Cognitive est un nouveau procédé et une nouvelle technologie qui permet le développement et l'amélioration de la Radio Logicielle, la relation entre ces deux technologies est montrée dans la figure I.1.[4]

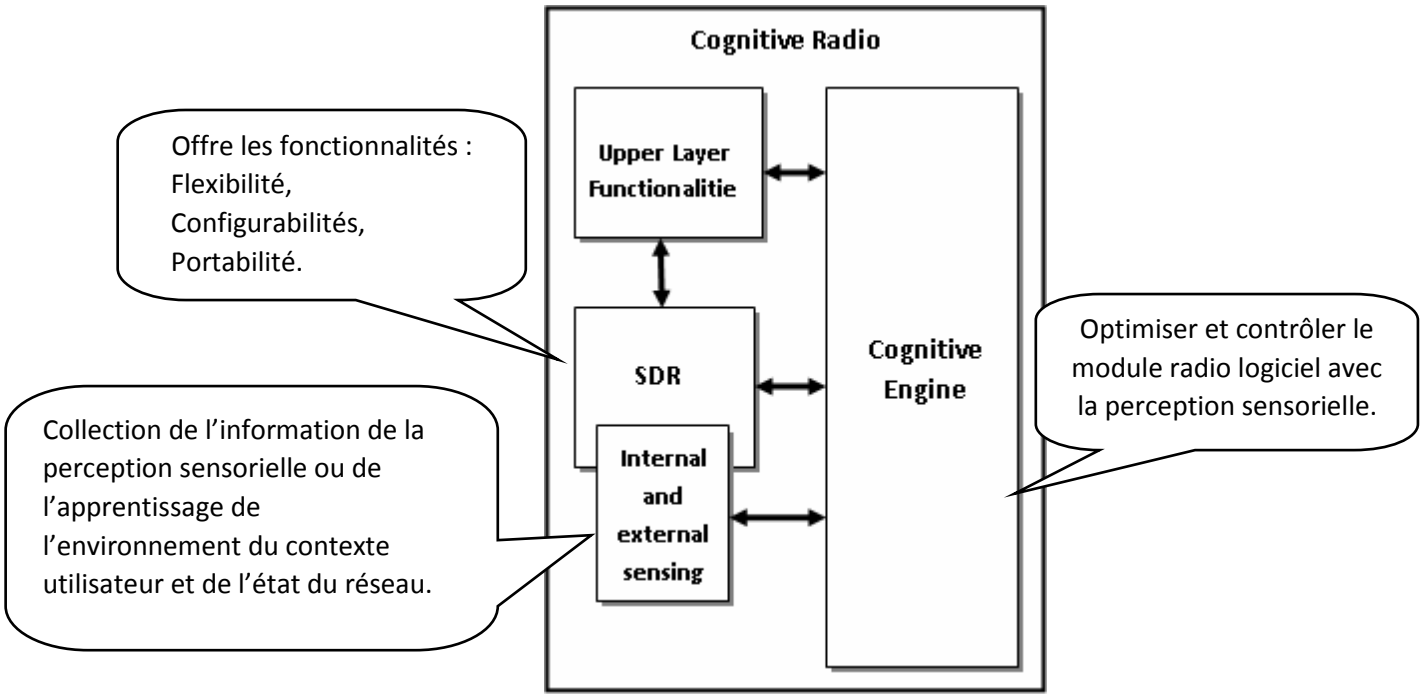

Figure I. 1 : La relation entre la RC et la SDR.

Cette technologie présente des avantages révolutionnaires, la RC permet d'avoir une optimisation d'utiliser des fréquences radio disponibles en changeant la fréquence d'opération quand cette dernière devient indisponible, de plus, elle autorise à tous les usagers un accès flexible, dynamique et de résoudre l'encombrement du spectre.

Selon Mitola : « Une radio cognitive peut connaitre, percevoir et apprendre de son environnement puis agir pour simplifier la vie de l'utilisateur »

Le principe de la radio cognitive, repris dans la norme IEEE 802.22, définie deux utilisateurs un dit primaires PU et c'est celui qui possède une licence sur une certaine bande de fréquence et un autre dit secondaire SU qui pourra à tous moments accéder à des bandes de fréquences qu'il a besoin (soit par la disponibilité, le débit …etc.) le SU se détache de ce contrat car il l'a loué, une fois le service est terminé ou bien le PU aura montré des velléités de connexion.

Afin d'avoir la collaboration et l'implication dans le même canal, une modification est faite sur la norme IEEE 802.16 (WiMax) en utilisant la norme 802.16h. La RC utilise les 2 normes pour la mise en œuvre. Le RRC (Réseau Radio Cognitif) combine les différentes bandes de fréquence et les différentes technologies en utilisant des bandes libres dans un temps fixe et un lieu donné.[5]

Les applications de la RC sont souvent incluses dans sa définition en raison du caractère impérieux et des applications uniques offertes. En outre, il existe de nombreuses techniques de radio logicielle que la RC est censée améliorer. Les éléments suivants sont souvent préconisés « applications de la radio cognitive » :

- Amélioration de l'efficacité spectrale.
- Amélioration de la fiabilité du lien radio.
- Topologie avancée du réseau.
- Techniques de collaboration.
- Automatisation de la gestion des ressources radio.[6]

I.3. Les fonctions de la Radio Cognitive

I.3.1. Détection du spectre

Permet de détecter les utilisateurs primaires et les espaces blancs du spectre, mesurer les interférences entre les PU et SU et avoir le statut du spectre (Libre/Occupé).

I.3.2. Gestion du spectre

Les fonctions de gestion du spectre sont importantes afin de répondre aux demandes de qualité de service sur les différentes bandes de fréquences. Ils sont placés pour :

- **Analyse du spectre :** déterminer la qualité c'est-à-dire mesurer par (signal /bruit, la durée moyenne et la disponibilité des espaces libres) en utilisant les algorithmes d'Intelligence Artificielle.
- **Décision sur le spectre :** la nécessité d'un modèle de décision pour l'accès au spectre, l'outil le plus utilisé pour la modélisation dans la RC est le processus de décision de Markov.

  Dans un système où il y a plusieurs utilisateurs l'accès au spectre doit être soit coopératif ou compétitif.

De plus, assurer de capter les meilleures bandes de fréquences pour répondre aux besoins des utilisateurs.

La figure I.2 présente les utilisateurs secondaires qui coopèrent entre eux pour accéder au spectre et maximisent une fonction commune, par contre, la figure I.3 montre que les utilisateurs entrent en compétition pour accéder au spectre radio.

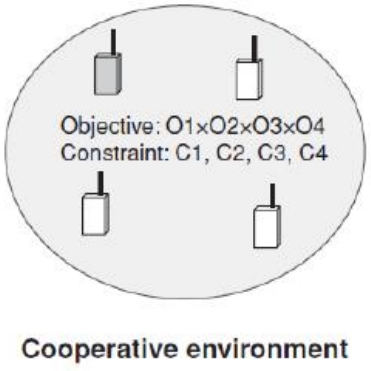

Figure I. 2: Accès Coopératif.

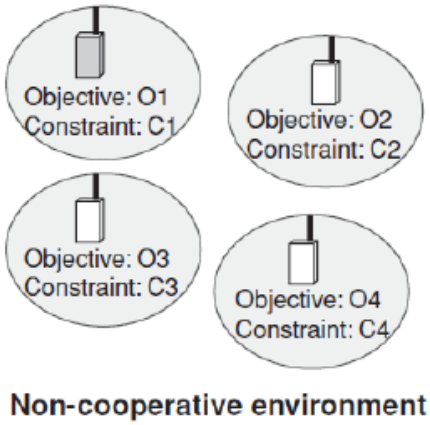

Figure I. 3: Accès Compétitif.

I.3.3. Mobilité du spectre

Permet la recherche des meilleures bandes de fréquences, auto-coexistence et synchronisation (la disponibilité du canal, la synchronisation du spectre).[7]

L'organigramme de la figure I.4 détaille les différentes fonctionnalités d'un nœud RC.[8]

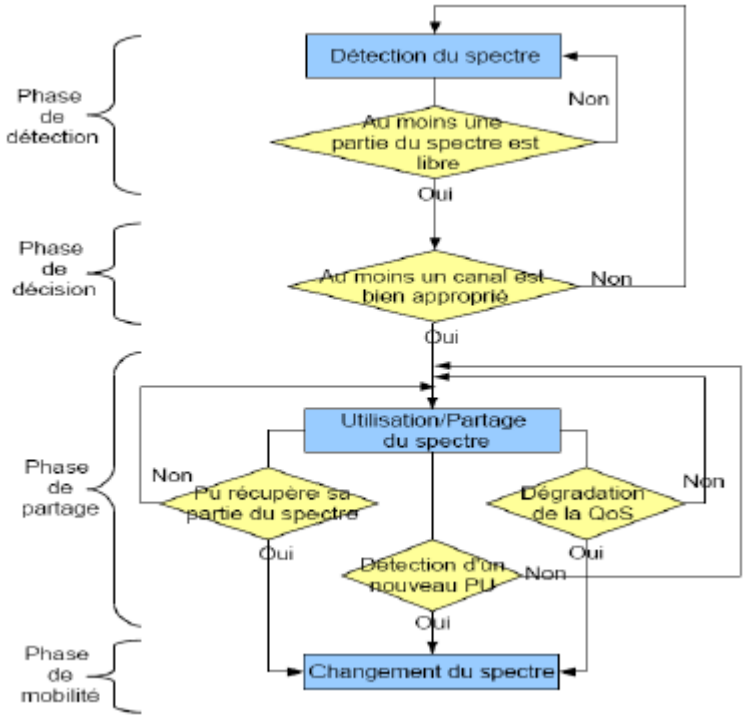

Figure I. 4 : Organigramme de fonctionnement d'un nœud radio cognitive.

I.4. Architecture de la RC

L'architecture de la RC est déterminée par Mitola, il utilise un ensemble de composants qui réalisent une série de fonctions de produits et de services, d'où la figure I.5 présente les composants du système de la Radio cognitive.

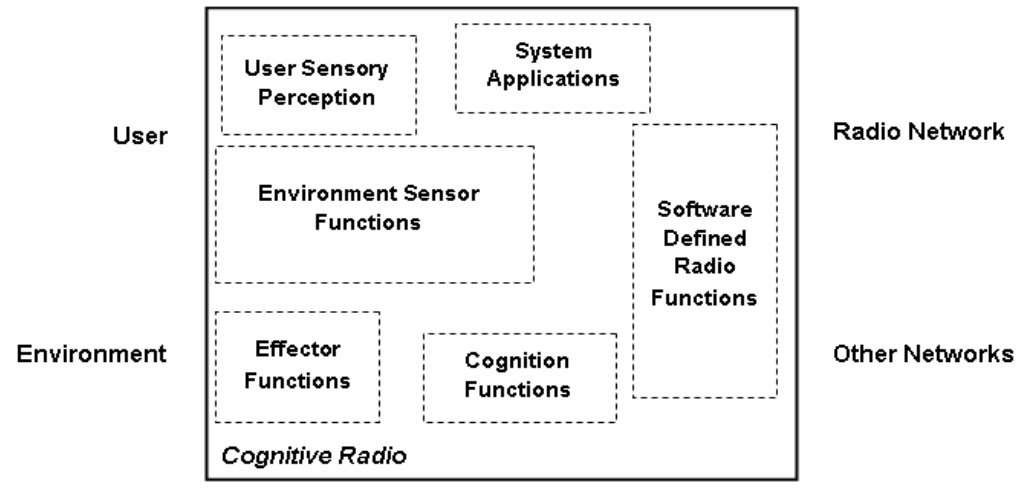

Figure I. 5: Architecture de la Radio Cognitive.

Les six composantes de l'utilisateur :

- User Sensory Perception : l'interface de l'utilisateur haptique (la vidéo, les fonctions de détection,)
- Enviroment Sensor Functions : l'emplacement, température …etc.
- System Applications : les services sont indépendants ex : jeux en ligne.
- SDR : détection de la FR, les applications radio de la SDR.
- Cognitive Functions : les systèmes de contrôle, planification.
- Effector Functions : les graphiques et les affiches multimédias…etc.[4]

La figure I.6 représente les différentes fonctionnalités et les protocoles utilisés par le RRC (Réseau Radio Cognitif) :

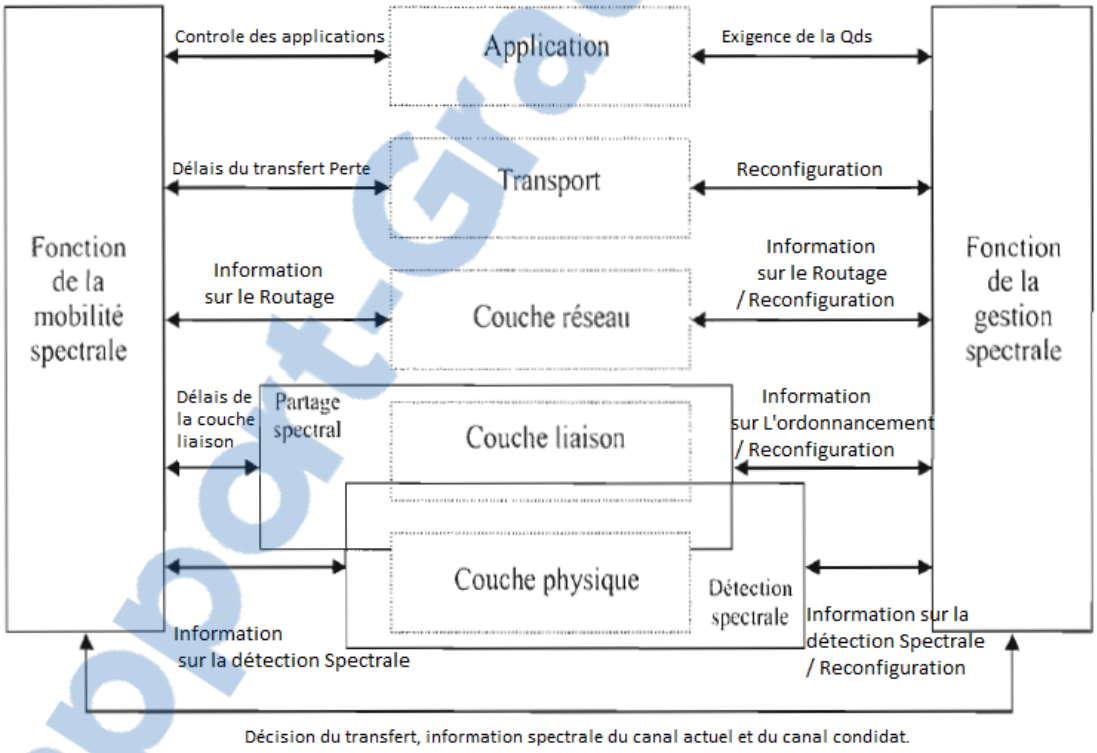

Figure I. 6: Fonctionnalités et protocoles utilisés par le Réseau Radio Cognitive.

Les variations de l'environnement RRC sont contrôlées par les différentes couches de communication, d'une part le partage et la détection du spectre se font par la coopération de la couche physique et la couche liaison, d'autre part la gestion et la mobilité exige un échange d'information de toutes les couches du modèle OSI[2].[1]

[2] OSI est un standard de communication, en réseau, de tous les systèmes informatiques. C'est un modèle de communications entre ordinateurs proposé par l'ISO qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

I.5. Cycle de cognition

La figure I.7 présente le système de fonctionnement de la Radio cognitive :

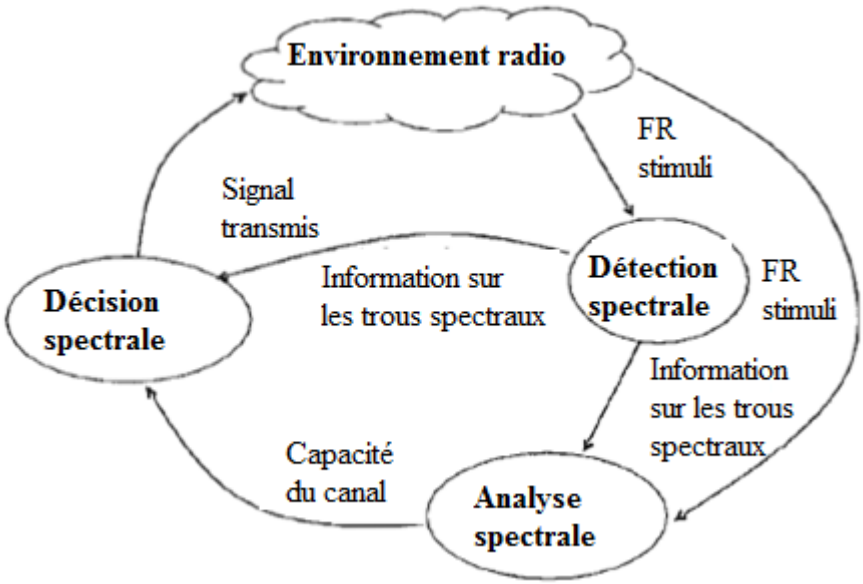

Figure I. 7: Cycle de cognition de Mitola

- **Détection spectrale**

La Radio Cognitive met les bandes spectrales qui sont à sa portée sous surveillance afin de capturer leur information et détecter les trous spectraux et la présence des utilisateurs qui opèrent sur les bandes licenciées.

- **Analyse de la bande spectrale**

Cette étape détermine l'analyse des caractéristiques des trous spectraux détectés, cette analyse détermine les meilleurs trous spectraux en termes de seuil d'interférences causés aux PU, la qualité des bandes de fréquences et la capacité du canal, la durée de disponibilité de la bande spectrale et finalement la position de l'émetteur par rapport au récepteur.

- **Décision spectrale**

Dans cette étape se fait le choix du débit nécessaire pour le transfert de données, le mode de transmission adéquat et la bande passante.[1]

I.6. Conclusion

Dans ce chapitre, nous avons présenté la technologie de la Radio Cognitive qui a pour but d'améliorer la fiabilité du lien sans fil pour un terminal radio cognitif mobile et d'assurer la flexibilité du canal en éliminant les interférences avec d'autres bandes fréquentielles. Ainsi les différentes fonctions et les protocoles qui sont utilisés par le Réseau Radio Cognitif.

Dans le deuxième chapitre, nous présentons les généralités sur la sécurité de l'information, les différentes attaques contre le Réseau Radio Cognitif et les solutions suggérées pour sécuriser ce réseau.

CHAPITRE II

Sécurisation du Réseau Radio Cognitif

II. CHAPITRE II : Sécurisation du Réseau Radio Cognitif

II.1. Introduction

La Radio Cognitive est un réseau d'une nouvelle génération qui se caractérise par son intelligence par rapport à ces ancêtres, son point fort c'est de pouvoir collecter les informations des réseaux entourés (WLAN, GSM/HSPA, LTE, WiMax, TV, Réseau Militaire) et utilise leur bande spectrale pour maximiser le débit des utilisateurs. L'apparition de cette technologie a pris aussi un coup d'œil sur le domaine de sécurité vue que ce réseau présente de nombreuses faiblesses en termes de sécurité.[9]

Dans ce chapitre, une explication sur la sécurité des attaques qui ciblent le RRC et les différentes propositions citées pour se défendre contre les menaces.

II.2. Fondamentaux sur la sécurité informatique

II.2.1. Définition

La sécurité informatique est un ensemble de stratégies bien organisées pour empêcher les utilisations non autorisées, modification et le divertissement du système d'information. La sécurité maintient la confiance des utilisateurs et des clients, l'objectif et d'accéder aux informations d'une manière légitime.

ISMS (Information security Management System) est un système de gestion de la sécurité de l'information, la figure suivante décrit les trois fondamentaux de la sécurité :[10]

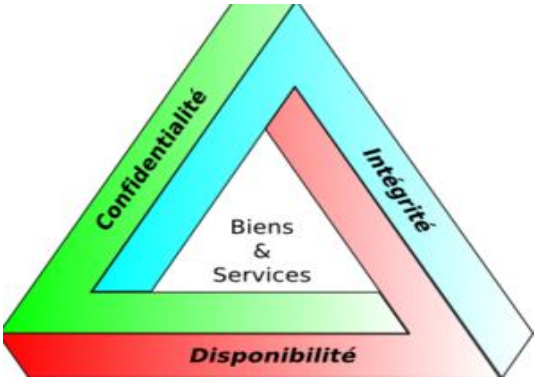

Figure II. 1: Triade des besoins fondamentaux de la sécurité.

- Disponibilité.
- Intégrité.
- Confidentialité.

II.2.2. Disponibilité

Permet d'assurer que le système est prêt à être utilisé, les ressources et les informations sont consommables. Elle assure aussi que les données ne soient pas saturées, et permet d'accéder aux systèmes d'une manière crédible. Les attaques visant la disponibilité tentent de rendre le système inexploitable et inutilisable.

II.2.3. Intégrité

Permet d'assurer que les données d'un système soient protégées contre toute modification, insertion, suppression ou relecture malveillante. L'intégrité est une assurance que les données reçues sont exactement telles qu'elles sont envoyées par une entité autorisée.

II.2.4. Confidentialité

La confidentialité est liée à l'intégrité et a pour objectifs d'empêcher les informations secrètes d'être révélées à des sujets non autorisés. Ainsi, d'assurer que seul les sujets autorisés auront accès aux ressources et aux informations, et pour assurer cela il faut faire appel à l'authentification.

II.2.5. Authentification

L'authentification permet de vérifier les informations d'une personne ou un processus informatique. C'est une preuve qui garantit que l'utilisateur est légitime et n'est pas un intrus. L'objectif principal d'un système d'authentification est d'empêcher les utilisateurs non autorisés d'accéder à des systèmes protégés. Il s'agit d'une procédure nécessaire pour vérifier l'identité et l'autorité d'une entité. Du point de vue du fournisseur de services, l'authentification protège le fournisseur de services contre les intrusions non autorisées dans le système. La plupart des mécanismes qui assurent l'authentification reposent sur une autorité de certification (CA) centralisée qui est approuvée par tous les utilisateurs du réseau. Un protocole d'authentification typique exigerait que les entités homologues obtiennent leurs identités signées (en utilisant le chiffrement de clé publique) par la CA et les certificats signés numériquement sont

échangés et vérifiés par les pairs afin d'assurer l'authenticité. Une fois l'authenticité des pairs établie, une communication régulière est initiée.[11]

L'IA (Information d'authentification) présente un des facteurs d'authentification selon le tableau suivant : [12]

Classification	Facteur	Exemple
Type 1 : Authentification par Reconnaissance.	Ce que vous savez.	• Mot de passe/passphrase. • Numéro d'identification personnel NIP
Type 2 : Authentification par possession.	Ce que vous possédez.	• Carte à puce. • Un jeton qui génère un mot de passe jetable (usage unique).
Type 3 : Authentification par caractéristique.	Ce que vous êtes.	• Empreinte digitale. • Reconnaissance d'iris.
Type 4 : Authentification par transmission.	Ce que vous transmettez	• Puce sous-cutanée dans le corps humain.

Tableau II. 1: les facteurs d'authentification.

II.3.Sécurité dans la Radio Cognitive

Il existe des solutions de sécurité de Réseau Radio Cognitif mais toute solution présentée contre les attaques RRC devrait respecter les exigences de la FCC « aucune modification du système historique ne devrait être nécessaire pour tenir compte de l'utilisation opportuniste du spectre par les utilisateurs secondaires » c'est-à-dire les solutions proposées doivent être implantées dans le système SU et non pas dans le système PU.[13]

Les attaques dans la RC ciblent les deux types radio :

Radio politique (Policy Radio) : permet de déterminer le comportement de la radio selon une certaine stratégie, ce qui fait que les informations de l'environnement seront transformées en statistique pour mesurer l'état de la radio.

Radio d'apprentissage (Learning Radio) : contiennent un moteur d'apprentissage qui définit une stratégie à base de ces connaissances, si ses informations changent la stratégie sera changée. Il permet de déterminer un bon fonctionnement des paramètres dans un environnement particulier, ce type de radio est connu pour être difficile contre les attaques.

La nécessité de citer les types de la RC permet d'établir les différentes attaques sur eux, par exemple dans la Policy Radio à l'aide de la connaissance de la manière de calcul des statistiques, l'attaquant peut attribuer et forcer une sortie souhaitée.[14]

Les attaques dans les RRC sont classées en fonction des couches OSI qu'elles ciblent : la couche physique, liaison, réseau et transport. Les attaques ciblent en particulier les réseaux Ad hoc et puisque les RRC sont considérés comme un type de réseau Ad hoc, cela implique que les attaques peuvent également se pointer contre un RRC.[9]

II.3.1. Les attaques de la couche physique (physical layer attacks)

II.3.1.1. Emulation de l'utilisateur Primaire (PUE)

Le mécanisme de la RC permet à un SU d'utiliser une bande de fréquence libre d'un PU. Après la détection de la BS (Bande Spectrale), le SU doit permuter les canaux vers la bande périodique pendant une durée d'allocation pour éviter les interférences avec le PU.

Dans le cas où un autre utilisateur secondaire utilise la même bande, une utilisation des dispositifs pour le partage du spectre d'une manière raisonnable est nécessaire.

L'attaque d'émulation de l'utilisateur primaire est composée de deux classes :

- L'attaque PUE égoïste : un SU malveillant émule un PU pour agrandir sa part du spectre. Elle peut être effectuée par deux attaquants pour mettre une liaison entre eux.

- L'attaque PUE malveillant : une attaque qui interdit les SU autorisés d'utiliser les espaces blancs d'un spectre.[13] [15] La figure au-dessous montre le mécanisme de l'attaque PUE :

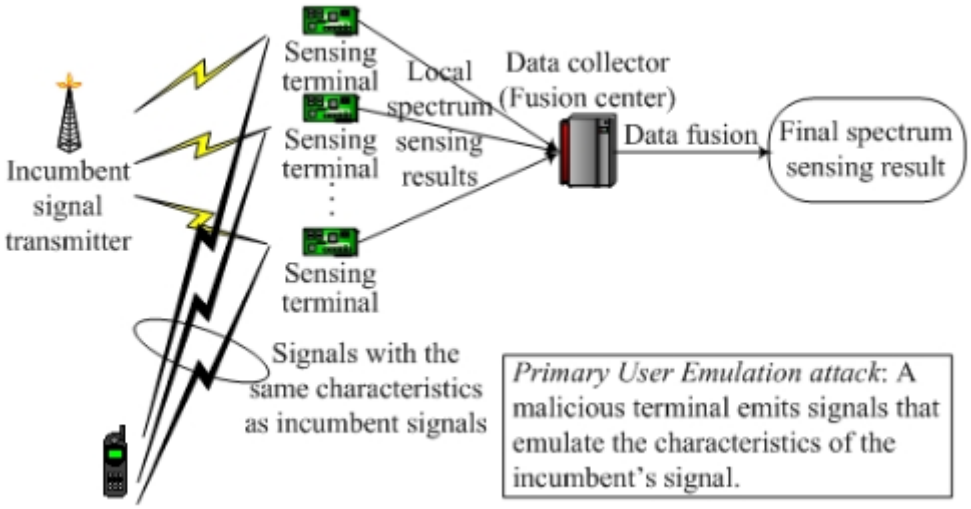

Figure II. 2 : L'attaque d'émulation de PUE.

Policy Radio et Learning Radio peuvent être provoquées par l'attaque d'émulation de l'utilisateur primaire. Ainsi que dans Policy Radio l'effet de l'attaque disparait aussitôt que l'attaquant quitte le canal[16], alors que dans Learning Radio l'attaquant collecte les informations du comportement de PU pour connaitre quand il va libérer le canal (temps mort), après il peut faire une attaque durant ce temps mort.

Les informations sur le RRC facilitent le fonctionnement des attaques PUE. Un exemple sur l'attaque PUE : l'attaquant profite du temps où les SU se refusent la transmission pour bien détecter le spectre. Un deuxième exemple quand le dispositif radio cognitive fait un changement de fréquence (Handoff), l'attaque peut être conduite à un faible débit ou entièrement à un DoS.[17]

- ✓ **Solution contre l'attaque PUE**

Pour se défendre contre l'attaque PUE, il faut d'abord connaitre la source du canal si c'est un PU ou bien un MSU (Malicious Secondary User) qui émule le PU.

L'identification d'un émetteur PU permet de séparer entre un PU légitime et un PU malveillant. La technique de l'authentification cryptographique permet la connaissance

de l'identité de PU, mais la restriction de la FCC interdit la modification du système PU. Donc, les chercheurs ont trouvé une solution efficace pour vérifier l'emplacement de la source PU c'est-à-dire une approche pour faire la correspondance entre l'emplacement de la source et l'emplacement de PU. [9]

Il existe deux techniques pour l'identification de l'emplacement de la source PU :

- **Test du rapport de distance (DRT)** : c'est-à-dire le calcul de la force de signal reçu.
- **Test de différence de distance (DDT) :** c'est-à-dire la différence de la phase du signal.

Les deux techniques utilisent une procédure de vérification de la source PU. L'objectif de cette procédure est de séparer entre les signaux primaires légitimes et les signaux malveillants.

Les tests de DRT et DDT sont effectués par des vérificateurs de localisation fiable LV qui sont classés en deux catégories :

- **Un maitre LV** : contient une Base De Donnée avec les coordonnées des tours de télévisions grâce à un système GPS[3].
- **Un esclave LV** : qui calcule la distance entre lui et le transmetteur par force du signal et le compare avec celui de la tour TV, les données ici doivent être cryptées et authentifiées pour ne pas les modifier ou les intercepter.

Les deux catégories sont liées pour contrôler leur communication. Une attaque de l'émetteur d'un signal est considérée si la vérification échoue.[13] Mais comme inconvénient à cette solution c'est que l'implémentation peut être couteuse de plus qu'elle peut être appliquée que dans un ad hoc à cause du mauvais signal émis.[18]

Le DRT et DDT peut être faux si un attaquant est prêt de la station TV comme solution à ce problème l'énergie des transmetteurs est la preuve de l'identité du transmetteur vue que celle du PU dépasse les 100 000 Kilo watt et moins de 1000 Kilo watt du MSU.

Les informations du niveau d'énergie sont très importantes pour l'attaquant, il doit les utiliser afin de dévoyer les SU.

[3] GPS : est un système de géolocalisation fonctionnant au niveau mondial et reposant sur l'exploitation de signaux radio émis par des satellites dédiés.

Une deuxième solution a été proposée pour se défendre contre l'attaque PUE c'est Localization Based Defense (LocDef) qui se résume en 3 étapes :

- Vérification des caractéristiques du signal
- Mesure du niveau d'énergie du signal reçu
- Localisation de la source du signal

Cette méthode utilise RSS-Based[4] localisation qui exploite la relation entre la force du signal et la position de l'utilisateur, quand la force du signal diminue cela veut dire que la distance entre l'émetteur et le récepteur est grande. Si un nœud assemble des données sur la puissance du signal à partir des nœuds distribués sur le réseau, il peut former un modèle de signal qu'il utilise pour connaitre la localisation de l'émetteur, et pour collecter les mesures RSS, un réseau capteur sous-jacent WSN (Wireless Sensor Network) est utilisé pour la collecte des mesures RSS. Un autre objectif pour WSN, il contribue à la détection du spectre et donne des informations sur les opportunités du réseau.[19]

L'empreinte digitale est la solution la plus efficace qui a été utilisée pour l'authentification,[20] au début, une approche qui permet d'améliorer la sécurité dans les réseaux sans fil a été proposée. La technique RFF utilise un procédé unique dans un temps court où l'émetteur est présent dans les ondes et activé. RFF est classé dans : système de synthèse de fréquence, sous-système modulateur, des amplificateurs RF et les propriétés physiques de l'émetteur. RFF permet la surveillance et l'analyse du signal analogique d'un réseau sur la couche physique. Donc, l'identification de l'émetteur et le problème de sécurité peuvent être résolues, mais cette solution n'est pas optimale à cause des calculs lourds et les gros échantillons. [21]

L'approche EMS (Signatures électromagnétiques) a été proposée pour résoudre le problème de l'optimisation. EMS permet la reconnaissance de motifs de signal de couche croisé. Le but de l'attaque PHY est de profiter de la simplicité, la flexibilité de la RC. EMS évite cette attaque car elle utilise l'identification des émetteurs, la détection, la collection des données et le test.

[4] RSS-Based : La localisation basée sur la puissance du signal reçu (RSS) est une méthode clé pour localiser les objets dans les réseaux de capteurs sans fil (WSN).

« EMS est un module de sécurité inter couche qui est capable de mettre en évidence les distinctions entre les dispositifs radio cognitifs. Il est conçu pour apprendre la caractéristique unique initiale à l'épreuve des dispositifs RC et le compare aux transmissions ultérieures pour validation et authentification »

EMS authentifie la source de l'émetteur et donc elle diminue les attaques DoS et les attaques PUE,[20] le fonctionnement de la technique de l'empreinte digitale est d'effacer la modulation des signaux reçus pour avoir un support avec bruit de phase. Les auteurs proposent l'empreinte digitale après une analyse et des statistiques logiques. Cette technique est la base de l'identification et contre les attaques PUE. [22]

II.3.1.2. L'attaque de la fonction objectif (Objective Function Attack)

« *Cognitive radio is a smart radio that has the ability to sense the external environment, learn from the history, and make intelligent decisions to adjust its transmission parameters according to the current state of the environment* »[23]

Le moteur cognitif adapte les paramètres radio telles que : la faible consommation d'énergie, le débit de données élevé, la haute sécurité, la fréquence centrale, la bande passante, la puissance, le type de cryptage …etc. Les fonctions objectives sont utilisées pour calculer ces paramètres. Par exemple : les paramètres radio qui maximisent le débit de données et minimisent la puissance. Le temps où le moteur cognitif est entrain de trouver les paramètres radio, un attaquant peut cibler ces paramètres pour que les résultats soient adaptés à son intérêt. Un exemple détaillé qui explique l'attaque de la fonction objective. Cette attaque n'affecte que le type Learning radio, le scénario de l'attaque est qu'à chaque fois que le moteur cognitif tente d'utiliser un niveau élevé de sécurité, l'attaquant lance un brouillage sur la radio en réduisant le taux de transmission R et réduisant aussi la fonction Objective $F = w1.R + w2.S$ où S taux de sécurité et w1,w2 représente les poids de R et S, Cette fonction objectif qui est utilisé par le moteur cognitif qui est responsable de l'ajustement des paramètres radio afin de répondre à des exigences spécifiques tel que la minimisation de la consommation d'énergie, le débit de données élevé et la haute sécurité. De cette façon l'attaquant force la radio à utiliser un niveau faible de sécurité.[16]

✓ **Solution contre l'attaque de la fonction Objective**

Que des suggestions ont été proposées pour se défendre contre l'attaquant parmi eux, définir des valeurs de seuil pour chaque paramètres radio, si les paramètres ne respectent pas les seuils la communication s'arrête, une autre suggestion a été présentée dans [17] c'est demander l'aide d'un système de détection d'intrusion IDS.

II.3.1.3. Jamming (L'attaque de Brouillage)

Le brouillage (Jamming) est une attaque qui cible les deux couches physique et mac, l'objectif de l'attaque est d'envoyer des paquets de manière continue à des utilisateurs légitimes afin de saturer la bande spectrale, reporter la transmission des SU et causer des interférences, aussi avoir une situation DoS (rendre le service indisponible, perturbation des communications…etc.). Un Jammer peut bloquer le canal qui fait les échanges entre les réseaux radio cognitifs. La connaissance et l'écoute des données de contrôle par l'attaquant est très dangereux pour le réseau radio cognitif.[14]

Le Jamming a quatre types de brouilleurs :

- **Jammer constant :** permet d'envoyer les paquets de données en continu sans faire une considération pour les protocoles de la couche Mac et sans attendre que le canal soit libre.
- **Jammer trompeur :** son but et de truquer les SU en envoyant des paquets de données excessive afin de les rendre en état reçu pendant une durée, il reste dans cet état lorsqu'il détecte un flux stable de données.
- **Jammer aléatoire :** il prend des pauses entre les signaux de brouillage, et il peut se comporter comme un Jammer constant ou trompeur.
- **Jammer réactif :** qui met le canal sous-surveillance et quand il détecte une communication sur le canal il commence le brouillage ce Jammer est le plus difficile à détecter car il ne transmet pas tout le temps.[24]

- ✓ **Solution contre l'attaque Jamming**

Vu que le DoS peut être appliqué dans les deux couches liaison et physique, chaque couche a sa méthode de détection :

Détection couche physique : les dispositifs légitimes utilisent une méthode de comparaison du bruit dans le réseau en recueillant suffisamment de données sur le niveau de bruit dans le canal et savoir si c'est normal ou anormal.

LCC (Location Consistency Checks) a été proposée pour la détection des brouillages dont l'emplacement est intéressant et qui est établi par GPS et informé par chaque nœud. Cette technique vérifie la cohérence des emplacements par exemple : un nœud est brouillé si ses voisins reçoivent un nombre minimal de paquets. La cohérence d'un PDR d'un nœud sera vérifiée avec ses voisins.

Détection couche Liaison : les dispositifs légitimes utilisent le protocole populaire d'accès au medium CSMA (Carrier Sensing Multiple Access), la détection d'un canal disponible sera faite par un périphérique, ce dernier ne transmet pas les données qu'après un délai de propagation. Si l'attaquant envoi les paquets et de manière continue le dispositif n'exécute jamais le protocole CSMA et il sera forcé de reculer, par conséquent le dispositif saura qu'il est victime d'un DoS.

Une autre technique a été proposée, son principe est d'analyser la relation entre la force du signal SS (Signal Strength) et le rapport de livraison des paquets PDR (Packet Delivery Ratio). PDR est le rapport des paquets livrés (nombre de paquets envoyés par un émetteur). Par exemple, dans le cas où SS est élevé et le PDR est faible l'utilisateur suppose qu'il est brouillé si ses voisins n'ont pas un SS et un PDR élevé.

Deux stratégies sont utilisées pour se préserver contre l'attaque Jamming (DoS) :

- Channel Surfing (déplacement des canaux ou changement de fréquence) : L'utilisation d'un canal différent lorsque l'attaque DoS est survenu.
- Spatial Retreat : L'emplacement des utilisateurs légitimes sera changé pour éviter les interférences de l'attaquant.[25]

II.3.2. Les attaques de la couche liaison (Link Layer Attack)

II.3.2.1. Falsification des données de détection du spectre

SSDF (Spectrum Sensing Data Falsification) ou l'attaque Byzantine, consiste à envoyer des données fausses sur la détection du spectre. La figure suivante montre le mécanisme de l'attaque Byzantine : [26]

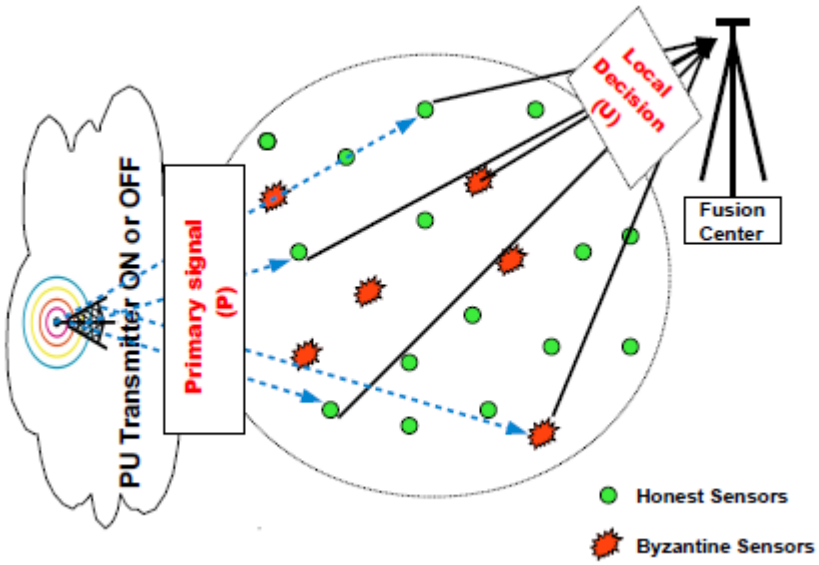

Figure II. 3: L'attaque Byzantine.

Cette attaque cible les réseaux radio cognitifs centralisés et distribués.

- RRC centralisé : la collection des données détectées et l'allocation des bandes de fréquences sont faites par un centre d'intégration.
  L'attaque SSDF (Byzantine) va tromper ce centre pour que les utilisateurs légitimes n'accèdent pas à des bandes de fréquences libre, ou ils peuvent accéder à des BS qui sont occupées.
- RRC distribué : la décision des bandes de fréquence est faite par la collaboration entre les réseaux radio cognitifs. SSDF est extrêmement malveillant dans les RRC distribués en raison de l'extension des données erronées.

Donc, dans le RRC centralisé l'effet des données malveillants est moins diminué car le centre d'intégration compare les données reçues de la RC avec quelques techniques intelligentes pour bien connaitre RC légitime. [27] [11]

Une analyse sur l'attaque byzantine a été faite où ils ont utilisé les limites de performance en termes de la fraction des attaquants byzantins quand aucune approche de défense sera fonctionnelle.[28]

Une étude a été faite pour bien analyser la performance du système RC en se basant sur la qualité de service (QoS) et la performance de détection des attaques et différentes contraintes sont traitées.[29]

- ✓ **Solution contre SSDF**

SDF (Several Data Fusion) sont des stratégies pour se défendre contre SSDF (Byzantine).

Parmi les techniques suggérées : la Décision Fusion, qui permettent de regrouper les données de la détection du spectre. Une condition sur l'addition a été faite : si la somme est supérieure ou égale au seuil, donc le résultat est occupé, un signal est présenté. Sinon le résultat indique que la bande est libre. A cause des interférences une stratégie qui supporte de prendre un et un seul seuil, dans ce cas, la détection sera erronée c'est-à-dire elle indique la présence d'un signal sur le réseau alors que ce n'est pas vrai, la bande est disponible.

L'attaquant SSDF profite de cette stratégie puisqu'elle indique toujours la présence d'un signal sortant et le résultat est toujours occupé, et pour résoudre ce problème, la valeur du seuil sera augmentée cela conduit à un accroissement de la probabilité de détection de défauts.[29]

WSRT (Weighted Sequential Ratio Test) est une stratégie qui a été suggérée pour se défendre contre les attaques SSDF.

Dans la structure Ad Hoc, les nœuds qui détectent le spectre vont rassembler les données et les rapports de détection des voisins.

Les deux étapes principales de cette technique sont :

- **Maintenance de la valeur :** chaque nœud a une valeur initiale égale à zéro, la valeur sera augmentée de 1 si le spectre est correct.[18]

- **Hypothèse d'essai de WSPRT** (Weighted Sequential Probability Ratio Test) : cette phase suppose le test de probabilité de séquence et la valeur du terminal. WSRT ressemble à la technique des réseaux de capteurs sans fils (WSN).[30]

Un dispositif a été suggéré pour l'identification des attaques Byzantins, permet de compter les décalages entre les décisions locales et les décisions globales, après la suppression des byzantins du processus. Cette technique est robuste contre les attaques SSDF.[31]

Une autre technique a été proposée, algorithme de détection des utilisateurs malveillants qui permet de calculer le niveau suspect des SU par une stratégie qui calcule une valeur de confiance c'est avec cette valeur que la séparation entre SU légitime et SU malveillant peut être faite. Tous ces systèmes de défense qui ont été cités ci-dessus ont des techniques et des mécanismes robustes et sécurisées, mais toujours la dégradation des performances.[32]

II.3.2.2. CCSD (Control Channel Saturation DoS Attack)

La négociation des canaux d'un processus RC est répartie, en utilisant un réseau radio cognitif multi hop. Pour la réservation du canal, des échanges de trame Mac seront faits dans cette étape de négociation. Dans le cas où tous les RRC communiquent en même temps, le canal supporte qu'un nombre limité des données, et donc l'attaquant profite de cette situation et il envoie des trames Mac truquées pour saturer le canal et diminuer les performances. Le fonctionnement de cette attaque est juste dans le RRC multi hop et non pas le centralisé car dans le centralisé les trames sont authentifiées par une station de base. [9]

II.3.2.3. SCN (Selfish Channel Negociation)

Le RRC peut rejeter la transmission des données pour d'autres réseaux, et alors le nœud RC peut conserver son énergie et augmenter le débit. Un scénario similaire lorsque l'hôte égoïste peut modifier le comportement Mac d'un RC. Cette attaque dégrade le débit du RRC.

- ✓ **Solution contre CCS et SCN**

Afin de minimiser la gravité de ces deux attaques CCS et SCN, il faut adapter une architecture de confiance où tout hôte suspect RC sera surveillé et évalué par ses

voisins. Un voisin peut alors effectuer une analyse séquentielle sur l'ensemble des données d'observation, et conclure une décision finale, qu'il s'agisse d'un mauvais comportement ou non. Le test de rapport de probabilité séquentiel peut être utilisé à cette fin, car il a prouvé son efficacité en termes de temps de détection.[33]

II.3.3. Les attaques de la couche réseau (Network Attack Layer)

Le développement dans le RRC s'est concentré sur les deux couches Physique et Liaison ce qui a causé des problèmes de routage, le RRC avec ces trois architectures présente des vulnérabilités même aux anciennes attaques du réseau sans fils. Dans ce qui suit une discussion sur les deux attaques les plus pertinentes contre le RRC, l'attaque Sinkhole (les puits) et l'attaque Hello Flood (inondation Hello).[27] [9]

II.3.3.1. Attaque Sinkhole

Dans cette attaque, l'attaquant se présente comme le meilleur itinéraire vers une destination spécifique attirant les nœuds voisins et transmettant leurs paquets ; cette attaque peut être la clé d'une autre attaque vue que les données pourront être lues, modifiées et supprimées. L'attaque n'est efficace que sur les architectures avec infrastructure et maillées où le trafic passe par une station de base.[9] [27]

- ✓ **Solution contre Sinkhole**

L'attaque de puits peut être difficile à détecter car elle exploite la même conception du protocole de routage et de l'architecture réseau, cependant il existe des protocoles qui ont empêché cette attaque comme le protocole Géographique, le principe de ce protocole est de construire une topologie aux besoins, en utilisant uniquement des communications et des informations locales sans avoir besoin d'initiation à partir de la station de base.[9]

II.3.3.2. Attaque Hello Flood

Cette attaque est plus défectueuse que celle décrite au-dessus, ici l'attaquant fait une diffusion à tous les nœuds du réseau avec une bonne qualité de service afin de les convaincre que c'est leurs voisins, par exemple un attaquant envoie un paquet publicitaire d'un lien de haute qualité vers une destination spécifique encouragera même les nœuds lointains à utiliser cette route et pourra les convaincre qu'il est leur voisin,

toutefois leurs paquets seront perdus et si un nœud découvre l'attaque il sera laissé sans voisin à transmettre ses paquets car tous vont utiliser la même malice route.[9]

- ✓ **Solution contre Hello Flood**

Pour se défendre contre cette attaque, l'idée c'est d'utiliser une clé symétrique, elle devrait être partagée avec une station base de confiance, la station de base servira de tierce partie de confiance comme dans Kerberos[5] et facilitera l'établissement des clés de session entre les partie réseaux. Afin de protéger leurs communication cette clé peut être utilisée par les nœuds pour vérifier l'identité de chacun et pour authentifier et chiffrer le lien entre eux, le nombre de clé partagé doit être limité pour empêcher n'importe quels nœuds intrus de créer une clé avec chaque nœud du réseau, de plus un nœud prétendant être le voisin de tant de nœuds dans un réseau doit déclencher une alarme, les algorithmes de clés symétrique sont les plus suggérés car ils sont rapides.

En général pour se défendre contre les attaques de routage, il y a des protocoles de sécurité de routage tel qu'un protocole de routage ad hoc SEAD (Secure Efficient Ad hoc Distance Vector) à vecteur de distance, ce protocole protège contre les attaques de DoS car il réalise une fonction de hachage unidirectionnel au lieu du cryptage asymétrique pour empêcher les attaquants de tenter de faire en sorte que d'autres nœuds utilisent plus de bande passante ou de temps de traitement.[27] [9]

II.3.4. Les attaques de la couche transport (Transport Attack Layer)

La couche transport (Transport Layer) peut être attaquée de plusieurs attaques qui visent les réseaux Ad Hoc sans fil, par exemple, l'attaque Lion qui cible le réseau radio cognitif.

L'attaque Lion peut être considérée comme une attaque cross-layer (multicouche) effectuée sur la couche physique qui applique PUE (l'attaque d'émulation d'utilisateur principal) afin d'interrompre la connexion TCP où les SU seront forcés de faire un changement de fréquence comme solution à l'attaque PUE, ainsi le protocole TCP ne

[5] Kerberos est un protocole d'authentification, son but est d'authentifier, d'autoriser et de surveiller les utilisateurs qui veulent accéder à des ressources ou des services du réseau. Il permet de résoudre les problèmes de sécurité, d'administration et de productivité dans l'authentification des services du réseau.

sera pas au courant de ce changement et continuera à créer des connexions logiques et à envoyer des paquets sans recevoir des acquittements, les segments TCP commencent alors à expirer et par conséquent TCP retransmet ces segments avec un timeout accrue et cela engendre une perte de paquets. De plus l'attaquant peut aussi interrompre les messages pendant le transfert de fréquence, cela conduit à une famine totale sur le réseau.[26]

- ✓ **Solution contre Lion Attack**

Pour diminuer l'effet de l'attaque du Lion, les auteurs Hernandez Serrano et AL proposent une stratégie qui permet de rendre le protocole TCP sensoriel de ce qui passe dans la couche physique, en utilisant le partage de données entre les couches : Physique, liaison et transport.[34]

Les mécanismes utilisés dans le RRC bloquent les paramètres de connexion TCP, pendant le changement de fréquence et les ajustent aux nouvelles conditions du réseau.

La gestion de clé de groupe GKM[6] (group key management) a été utilisée pour que les données de contrôle seront fiables. Cette technique permet de crypter, décrypter et d'authentifier les membres du réseau radio cognitif. Aussi, un IDS (Système de Détection d'Intrusion) multicouches a été utilisé pour trouver la source d'attaque, comme indiqué dans la figure suivante :[26]

[6] GKM est Group Key Management signifie gérer les clés dans une communication de groupe. La plupart des communications de groupe utilisent la communication multidiffusion de sorte que si le message est envoyé une fois par l'expéditeur, il sera reçu par tous les utilisateurs. Le principal problème dans la communication de groupe multicast est sa sécurité. Afin d'améliorer la sécurité, différentes clés sont données aux utilisateurs. En utilisant les touches, les utilisateurs peuvent crypter leurs messages et les envoyer en secret.

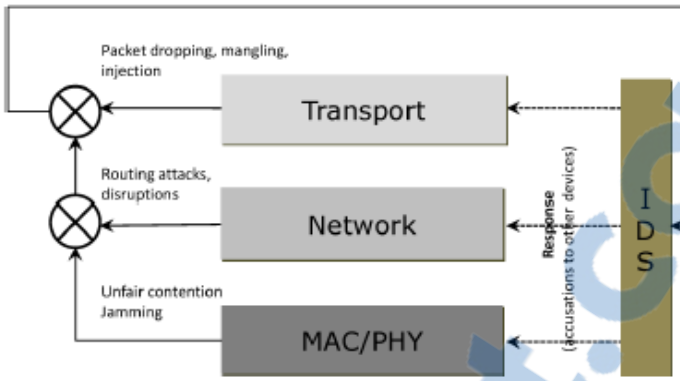

Figure II. 4: IDS, Système de détection d'intrusion.

II.4. Conclusion

Ce chapitre traite la sécurisation des Réseaux Radio Cognitifs et nous avons présenté les différentes attaques possibles contre le RRC selon le classement des couches du modèle OSI et les différentes solutions et les propositions suggérées.

Donc, le RRC est construit sur la base des technologies existantes, les mécanismes et les techniques afin d'assurer une forte sécurité. Mais cela ne suffit pas car le RRC a des caractéristiques particulières et en compte tenu des limites imposées par la FCC, alors il y a toujours de nouvelles attaques qui apparaissent et aussi des nouvelles propositions de sécurité sont nécessaire.

Tout au long du chapitre, nous avons établi les menaces pour les différentes couches : brouillage, falsification, routage…etc. De plus, nous avons cité des mesures de sécurité pour protéger le réseau radio cognitif. Au fur et à mesure que les RRC continueront de se développer et de devenir plus courant, d'autres stratégies de sécurité seront nécessaires. Surtout, l'authentification des signaux et les techniques qui permettent de détecter les données malveillantes vont vaincre la plupart des attaques spécifiques au RRC.

Chapitre III

Contribution et résultat

III. CHAPITRE III : Contribution et résultats

III.1. Introduction

Par rapport aux réseaux sans fil classiques, les réseaux RC sont en outre soumis à une émulation d'utilisateurs autorisés et à des attaques contre les gestionnaires du spectre, à moins que des mécanismes de sécurité robustes ne soient mis en place. L'un des types les plus courants d'attaques dans les réseaux RC est l'attaque d'émulation de l'utilisateur primaire (PUE) qui pourrait affecter le réseau radio cognitif.[35]

Dans ce chapitre, nous avons traité le cas de l'attaque PUE qui permet d'imiter les caractéristiques des signaux électriques d'un PU légitime et faire en sorte que les SU légitimes identifient d'une façon erronée. Cela peut être réalisé en imitant les critères des transmissions PU. Dans cette attaque, l'adversaire doit convaincre les SU que le signal émulé provient d'un PU authentique et pour solution les chercheurs ont trouvé une solution efficace pour vérifier l'emplacement de la source PU c'est-à-dire une approche pour faire la correspondance entre l'emplacement de la source et l'emplacement du PU.[36]

D'autres solutions ont été suggérées pour la sécurité et l'authentification des nœuds RC qui peuvent être obtenues grâce à des techniques cryptographiques tel que le protocole d'authentification EAP.[37]

Aussi l'attaque de Hello Flood qui permet d'envoyer plusieurs messages Hello afin d'inonder le réseau et pour solution contre cette attaque, l'utilisation d'une clé symétrique pour l'identification et l'authentification.

Dans ce contexte, nous avons pris l'idée de l'attaque d'émulation des PU et l'attaque Hello Flood pour instaurer notre algorithme dans le contexte d'un réseau radio cognitif.

Avant de commencer cette partie, nous allons décrire le cas où il y a plusieurs CPU contre un seul CSU. Nous précisons que la simulation est développée avec le langage JAVA sous l'environnement de développement intégré Netbeans.

Notre travail consiste à sécuriser la communication entre les CPU et le CSU contre les deux attaques citées au-dessus afin d'avoir une meilleure utilisation du spectre sans

interférences. Pour réaliser cette étude nous avons implémenté un algorithme qui utilise les méthodes de la décision multicritère qui est TOPSIS et un algorithme de clé symétrique Blowfish.

III.2. Suppositions

Nous rappelons que notre système contient un CSU et plusieurs CPU. Dans ce qui suit et pout pouvoir proposer un algorithme qui respecte les normes, nous supposons que :

- Chaque utilisateur est doté d'un agent.
- Le PU est en position fixe à t_0 (pas de mobilité).
- La couche physique est sécurisée car selon la FCC, les infrastructures radio doivent maintenir la sécurité physique des utilisateurs.[38]
- Les liens entre les terminaux sont fiables.
- Le comportement d'un PUE est toujours mensonger, il essaye de mimer le comportement d'un PU et essaye toujours de proposer les meilleures offres afin que les SU partagent avec lui le spectre non utilisé.
- Le nœud malicieux se trouve dans la même zone géographique que l'agent de coalition.
- Les agents de coalitions CPU sont toujours honnêtes.
- Chaque utilisateur connait sa propre clé pour pouvoir s'authentifier.
- Pour que l'authentification se fasse correctement, nous supposons que les nœuds peuvent faire une synchronisation en temps réel afin de connaitre les clés publiques des différents nœuds en utilisant n'importe quel protocole standardisé.

Dans ce mémoire, nous ne traitons que les attaques de type PUE et Hello Flood. Nous ne nous intéressons pas aux autres types d'attaques.

III.3. Les systèmes multi-agents

L'apparition des Systèmes Multi-Agent (SMA) a été au carrefour des recherches sur l'intelligence artificielle distribuée et sur la vie artificielle. Les SMA sont utilisés pour proposer des solutions réactives et robustes à des problèmes complexes où il n'y a pas un contrôle centralisé.

Un SMA est un ensemble d'agent qui travaillent ensemble afin de résoudre un problème ou effectuer une activité. [39]

Ferber [40] définit un SMA de la manière suivante :

« Un système Multi-agents est un système composé des éléments suivants :

- ✓ Un **environnement** est un espace disposant généralement d'une métrique.
- ✓ Un ensemble d'**objets** situés dans l'espace, ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents.
- ✓ Un ensemble d'**agents** qui sont les entités actives du système.
- ✓ Un ensemble de **relations** qui unissent les objets entre eux.
- ✓ Un ensemble d'**opérations** permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets.
- ✓ Un ensemble d'**opérateurs** chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers). »

III.4. Outils utilisés

III.4.1. JADE

JADE est un système qui est implémenté avec le langage Java. JADE est un middleware qui simplifie la mise en œuvre des systèmes multi-agents. [41]

C'est une plateforme agent qui satisfait aux spécifications de la FIPA [5], et un API pour développer des agents en Java. JADE est contrôlée par Telecom Italia Lab. Cette plateforme est composée de conteneurs actifs :

- **Conteneur :** c'est un environnement qui permet l'exécution concurrente de plusieurs agents, aussi il contrôle le cycle de vie des agents et il assure la communication entre les agents.

- **Conteneur principale (Main Container) :** permet d'héberger l'AMS et le DF.
  - **AMS (Agent Management System) :** référence l'identité des agents (leur nom) dès leur entrée dans le système.
  - **DF (Directory Facilitator) :** c'est un service qui permet de référencier les demandes des agents selon leur service. [42]

La figure ci-dessous montre le fonctionnement de la plateforme JADE :

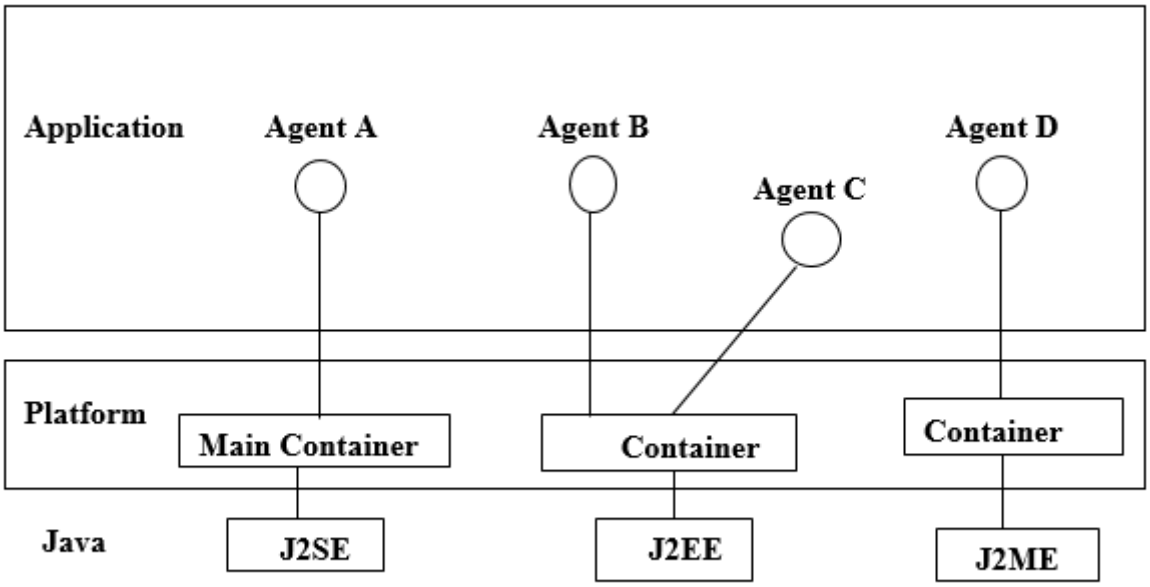

Figure III. 1: La plateforme JADE

❖ **Communication entre agents et FIPA ACL**

« FIPA ACL (FIPA Agent Communication Language) est un langage de communication entre agents créé par FIPA dont la spécification consiste en un ensemble de types de messages et en un ensemble de protocoles d'interaction de haut niveau. Le but de FIPA ACL est d'interagir entre les agents quelques soit le protocole qu'ils utilisent.

Les actes de communication sont accomplis à travers l'envoi de messages d'un agent à un autre en utilisant les spécifications établies par la FIPA. Un message FIPA ACL contient un ensemble de paramètres. Le seul paramètre obligatoire est la

«performative ». Néanmoins, la plupart des messages doivent contenir des paramètres tels que l'expéditeur, le destinataire et le contenu. » [39]

Le tableau qui va suivre nous montre les différentes performatives que nous avons utilisé :

Inform	Communication par l'expéditeur d'une proposition, pensée vraie par celui-ci.
Request	Communication par l'expéditeur d'une demande au destinataire d'effectuer une action.
Propose	Communication par l'expéditeur d'une proposition d'action conditionnée à certaines pré conditions données.
Accept-proposal	Communication de l'accord de l'expéditeur d'effectuer une action qui lui a été préalablement soumise.
Not-understood	Communication par l'expéditeur d'une non compréhension d'une action effectuée par le destinataire.
Subscribe	Communication par l'expéditeur d'une demande d'un objet donnée par une référence envoyé par l'expéditeur, et de renotifier l'agent ayant souscrit dès que l'objet en question change.

Tableau III. 1: Actes de communication du modèle FIPA ACL

III.4.2.SQLite

SQLite est une bibliothèque en cours qui implémente un moteur de base de données SQL transactionnel indépendant, sans serveur, à configuration nulle et transactionnel. C'est la base de données unique, qui est configurée à zéro, ce qui signifie que c'est une base de données dont vous n'avez pas besoin de la configurer dans votre système.

Le moteur SQLite n'est pas un processus autonome comme d'autres bases de données, vous pouvez le lier de manière statique ou dynamique selon vos besoins avec votre application. SQLite accède directement à ses fichiers de stockage.[43].

III.4.3.Netbeans

Netbeans est un IDE qui offre un meilleur support pour le développement des applications web et le coté serveur qui utilise la plateforme Java EE. Cet IDE a été développé en étroite collaboration avec les équipes Java EE et Glass Fish pour fournir

une intégration et une utilisation plus simple de la spécification Java EE. L'utilisation de Netbeans est un moyen d'apprendre rapidement et de devenir productif dans la programmation Java EE.[44]

III.5. Contribution

III.5.1.Scénario

Notre étude concerne un seul CSU avec plusieurs CPU. A l'aide de l'algorithme TOPSIS [Annexe A], le CSU et les CPU négocient leur accord selon des critères tels que le prix, le nombre de canaux et le temps d'allocation.

La figure suivante montre les messages échangés entre CSU et les CPU. Tout d'abord, le CSU contacte les CPU pour signaler le nombre de canaux dont il a besoin. Après, chaque CPU propose son offre. Puis, le CSU reçoit les offres des CPU et en appliquant l'algorithme TOPSIS, il choisit la meilleure offre. Et pour que la communication entre le CSU et les CPU soit sécurisée, il faut que le choix de la meilleure offre soit sécurisé également, c'est dans ce contexte que nous avons instauré notre algorithme qui sera détaillé dans ce qui suit.

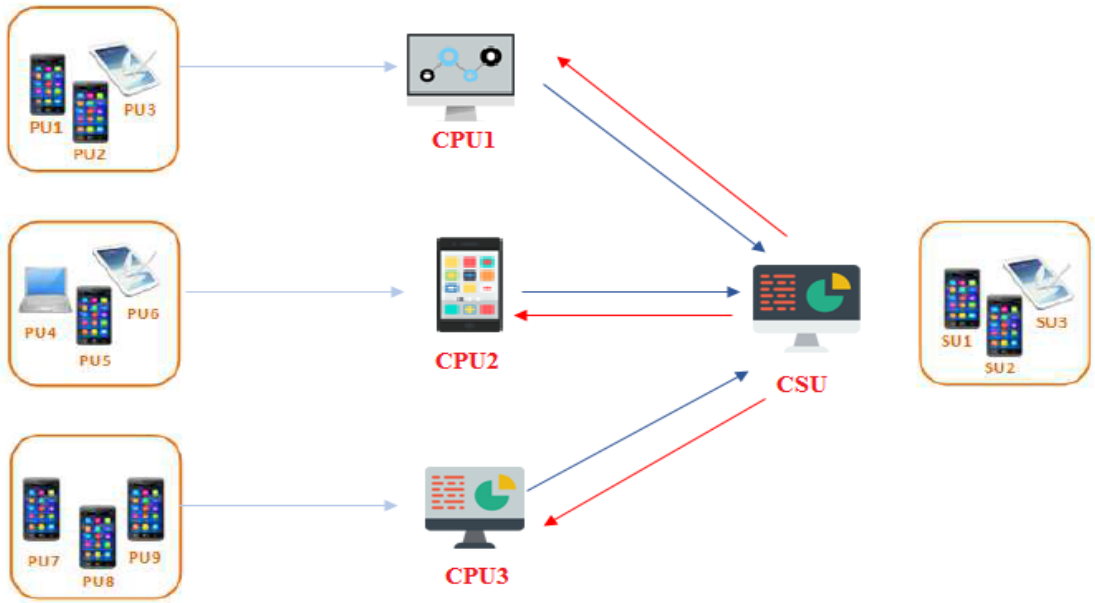

Figure III. 2 : Scénario proposé.

III.5.2.Travail effectué

III.5.2.1. Coté CPU

La figure III.3 décrit le comportement du CPU lors de l'application de l'algorithme, le CPU envoie un message **INFORM** au CSU pour le renseigner qu'il est présent. Le CSU envoi sa requête au CPU. Si le CPU peut satisfaire la demande, il propose l'offre avec le nombre de canaux, le prix et la durée. Ensuite, si le CPU reçoit un message fictif, il envoie une réponse de Type **NOT_UNDERSTOOD** pour lui dire qu'il n'a pas compris la requête, dans ce cas, le CSU saura que le CPU n'est pas malveillant et lui enverra un message **ACCEPT_PROPOSAL** pour lui dire qu'il a accepté sa proposition et fait l'authentification avec un message **SUBSCRIBED** et établie une connexion. Sinon, il reçoit l'accord avec un message **ACCEPT_PROPOSAL**, CPU authentifie avec un message **SUBSCRIBED** qui a comme paramètre son Id et une clé symétrique.

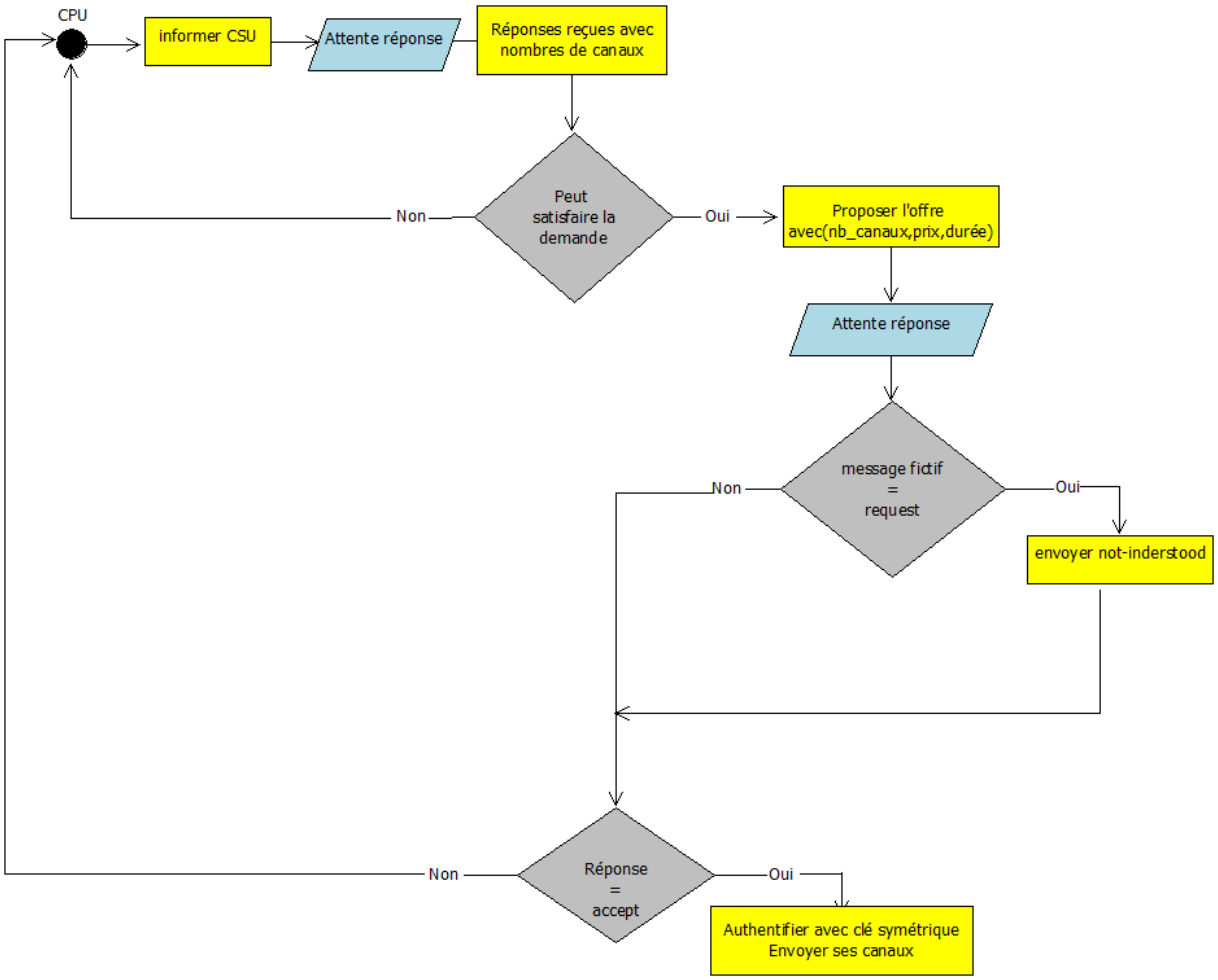

Figure III. 3 : Comportement du CPU.

III.5.2.2. Coté CSU

III.5.2.2.1. Simulation de l'apprentissage

L'apprentissage est fait par un CSU où les CPU voisins envoient des messages **INFORM** avec leur Id et une clé symétrique (générée par Blowfish) et sont enregistrés dans une base de données Agent_CPU comme le montre la figure III.4.

Tous les CPU arrivés dans cet intervalle de temps sont considérés (supposition) comme fiables. Ceci dit, si un message **INFORM** arrive d'un autre CPU après l'expiration du temps, il ne sera pas enregistré dans la base de données mais il sera pris en compte.

	Id_CPU	Clé_sym
	Filter	Filter
1	CPU0	011110010110010100100110000111101100100101011...
2	CPU1	101111110110101100111000111110101011000110001...
3	CPU2	100100101011001010100110101011000111110011100...

Figure III. 4: Base de données Agent_CPU.

III.5.2.2.2. Fonctionnement de l'Algorithme

A un certain moment, le CSU aura besoin d'un nombre de canaux donc il va envoyer une requête à tous les CPU qui ont envoyé un message de type **INFORM**, alors les CPU qui peuvent satisfaire cette offre vont répondre par des propositions avec un message **PROPOSE** qui contient certains critères (Nombre de canaux, Prix, Durée), ces critères vont être enregistrés dans une autre base de données Historique comme le montre la figure III.5 qui sera utilisée plus tard. Pour choisir la meilleure offre nous avons utilisé l'Algorithme TOPSIS.

	Id	Cannaux	Prix	Temps	Topsis
	Filter	Filter	Filter	Filter	Filter
1	636	7	2	6	0.700132076208...
2	647	6	1	4	0.68068757136597
3	688	7	1	4	0.679402942627...
4	714	6	2	8	0.714047417332...
5	743	5	1	8	0.693759901174...

Figure III. 5: Base de données Historique.

Nous rappelons que dans notre travail, nous avons traité deux types d'attaques (Hello Flood et PUE). Alors pour Hello Flood, nous avons ajouté un test au niveau de notre algorithme qui consiste à compter le nombre de messages **INFORM** reçus de la part d'un même utilisateur.

Si un CPU envoie plusieurs messages **INFORM** pour signaler sa présence, il va saturer le réseau et donc sera considéré automatiquement comme malveillant.

Concernant, l'attaque PUE nous avons proposé deux méthodes :

a. Méthode Optimal CR

Cet Algorithme utilise la fonction MacPropo qui trie les résultats obtenus du TOPSIS par ordre décroissant et qui est détaillée au-dessous. Si le CPU qui possède le meilleur choix appartient à la base de données, il sera considéré comme fiable. Sinon, le CSU compare la valeur TOPSIS de ce CPU avec la moyenne des résultats TOPSIS qui ont été enregistrés dans la table Historique et qui est calculée par :

$$\boldsymbol{moyTopsis = (max + \mathbf{min})/2}$$

max : la valeur maximale du résultat TOPSIS qui est déjà stockée dans une base de données.

min : la valeur minimale du résultat TOPSIS qui est déjà stockée dans une base de données.

Puis, si le résultat TOPSIS est supérieur à la moyenne donc le CPU est suspect d'être malveillant et on refait l'appel à cet algorithme. Sinon, un message fictif sera envoyé, s'il y a une réponse de la part de ce CPU de type **NOT_UNDERSTOOD** dans un intervalle de temps de (2000 ms) le CPU sera considéré comme fiable et il reste l'authentification et le renvoi des résultats au CPU choisi. Sinon le CPU sera suspect d'être malveillant et on refait l'appel à cet algorithme.

La figure III.6 montre l'algorithme Optimal CR que nous avons proposé pour sécuriser la communication entre CPU et CSU.

Algorithm 1 Algorithm Optimal CR

```
1: CPU envoie inform
2: if id inform n'est pas recu then
3:     CSU envoi une requete au CPU
4:     if plusieurs offres then
5:         appliquer TOPSIS
6:         appel fonction MacPropo
7:         if CPU appartient base de donne then
8:             CPU fiable,accepter
9:             envoi cle symetrique, "authentification"
10:        else
11:            moyTOPSIS = (max + min)/2
12:            if resTOPSIS > moyTOPSIS then
13:                CPU est suspect d'etre malveillant
14:                appel la methode Optimal CR
15:            else
16:                envoyer message fictif = request
17:                Attente reponse pendant une duree
18:                if reponse = not inderstood then
19:                    CPU fiable
20:                    envoi cle symetrique, "authentification"
21:                else
22:                    CPU est suspect d'etre malveillant
23:                end if
24:            end if
25:        end if
26:    end if
27: else
28:    CPU est suspect d'etre malveillant
29: end if
```

Figure III. 6: Algorithme Optimal CR.

- **Méthode MacPropo :** le type de retour entier et qui a comme paramètre (Double ResTopsis [], int i), cette méthode a comme objectif de trier le tableau ResTopsis qui contient les résultats de l'Algorithme TOPSIS reçu comme paramètre en ordre DSC et retourne la position du meilleur résultat demandé dans ResTopsis [] avant qu'il soit trié.

La figure suivante montre l'implémentation de la fonction MacPropo :

Algorithm 1 Algorithme Proposition d'offre maximum

```
1:  function MACPROPO(tab,indice)
2:      re ← 0
3:      tab1 ← tab.length
4:      TAB ← tab1.length[2]
5:      for j ← 0 to tab.length do
6:          tab1[j] ← tab[j]
7:          TAB[j][0] ← tab[j]
8:      end for
9:      for i ← 0 to tab1.length − 1 do
10:         index ← i
11:         for j ← i + 1 to tab1.length do
12:             if tab1[j] > tab1[index] then
13:                 index ← j
14:             end if
15:         end for
16:         smallerNumber ← tab1[index]
17:         tab1[index] ← tab1[i]
18:         TAB[index][0] ← tab1[i]
19:         tab1[i] ← smallerNumber
20:         TAB[i][0] ← tab1[i]
21:     end for
22:     for i ← 0 to tab.length do
23:         for j ← 0 to tab.length do
24:             if TAB[i][0] = tab[j] then
25:                 TAB[i][1] ← j
26:             end if
27:         end for
28:     end for
29:     re ← TAB[indice][1] return re
30: end function
```

Figure III. 7: Algorithme du choix d'offre maximum.

La figure III.8 décrit le comportement du CSU utilisé lors de l'application de l'algorithme Optimal CR :

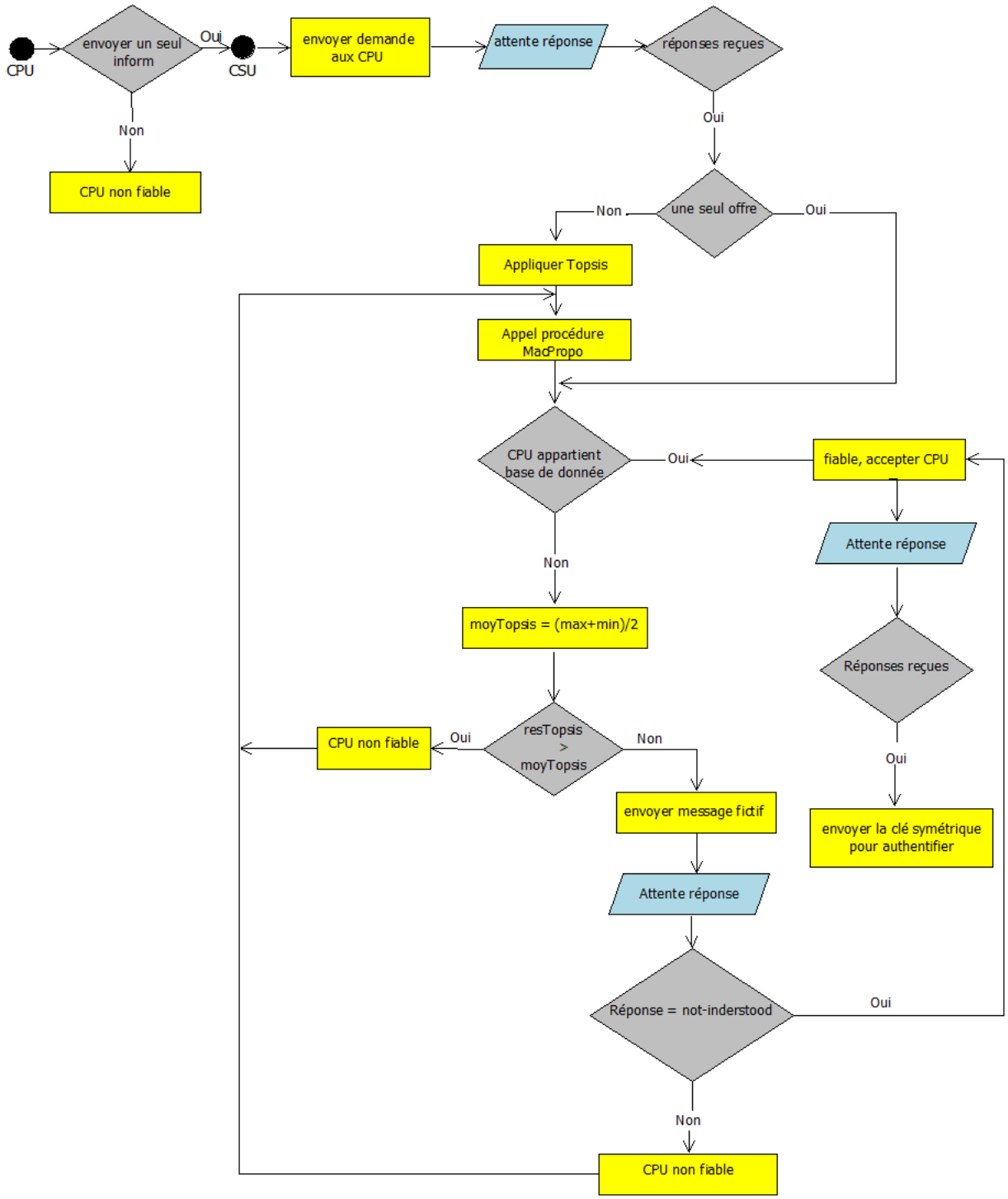

Figure III. 8 : Comportement du CSU lors de l'application de l'algorithme Optimal CR.

La figure ci-dessous montre le résultat des différentes interactions possibles entre les utilisateurs lors de l'application de l'algorithme Optimal CR.

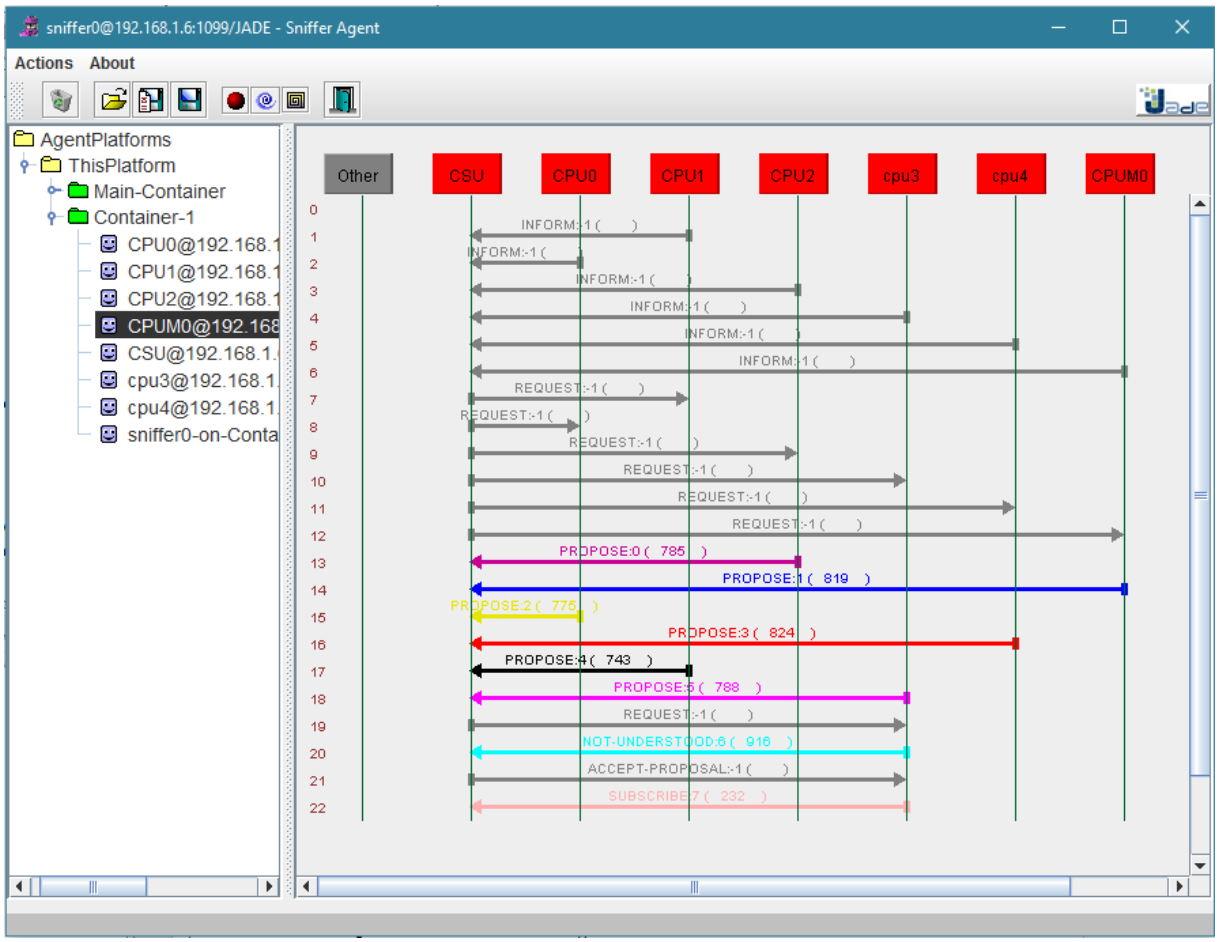

Figure III. 9: Agent Sniffer pour l'algorithme Optimal CR.

b. Méthode Secure CR

Cet algorithme suggéré utilise le même principe de l'algorithme précédant. La différence réside après l'application du TOPSIS et le calcul de la moyenne TOPSIS. Nous avons proposé de soustraire le résultat TOPSIS et la moyenne TOPSIS comme suit :

$$\boldsymbol{moyTopsis = (max + min)/2}$$

$$\boldsymbol{resTopsis = resTopsis - moyTopsis}$$

Puis, le CSU appelle la fonction MicPropo qui trie les ResTopsis obtenus par l'équation décrite au-dessus par ordre croissant et qui est détaillée par-là suite, Si le CPU qui possède la meilleure offre appartient à la base de données, il sera considéré comme fiable. Sinon, un message fictif sera envoyé, s'il y a une réponse de la part de ce CPU de type **NOT_UNDERSTOOD** dans un intervalle de temps de (2000 ms) le CPU sera considéré comme fiable et il reste l'authentification et le renvoie des résultats au CPU choisi. Sinon le CPU sera suspect d'être malveillant et on refait l'appel à cette méthode.

La figure suivante montre l'algorithme Secure CR que nous avons suggéré :

Algorithm 1 Algorithm Secure CR

```
1: CPU envoie inform
2: if id inform n'est pas recu then
3:     CSU envoi une requete au CPU
4:     if plusieurs offres then
5:         appliquer TOPSIS
6:         moyTOPSIS = (max + min)/2
7:         for i ← 0 to resTOPSIS.length do
8:             resTOPSIS [i] = resTOPSIS[i] - moyTOPSIS
9:         end for
10:        appel fonction MicPropo
11:        if CPU appartient base de donne then
12:            CPU fiable,accepter
13:            envoi cle symetrique, "authentification"
14:        else
15:            envoyer message fictif = request
16:            Attente reponse pendant une duree
17:            if reponse = not inderstood then
18:                CPU fiable
19:                envoi cle symetrique, "authentification"
20:            else
21:                CPU est suspect d'etre malveillant
22:            end if
23:        end if
24:    end if
25: else
26:    CPU est malveillant
27: end if
```

Figure III. 10: Algorithme Secure CR.

- **Méthode MicPropo :** le type de retour est entier et qui a comme paramètre (Double ResTopsis [], int i), cette méthode a comme objectif de trier le tableau ResTopsis qui contient la différence des résultats TOPSIS avec la moyenne reçue comme paramètre en ordre ASC et retourne la position du meilleur résultat demandé dans ResTopsis [] avant qu'il soit trié.

La figure ci-dessous montre l'implémentation de la fonction MicPropo :

Algorithm 1 Algorithme Proposition d'offre minimum

```
1:  function MICPROPO(tab,indice)
2:      re ← 0
3:      tab1 ← tab.length
4:      TAB ← tab1.length[2]
5:      for j ← 0 to tab.length do
6:          tab1[j] ← tab[j]
7:          TAB[j][0] ← tab[j]
8:      end for
9:      for i ← 0 to tab1.length − 1 do
10:         index ← i
11:         for j ← i + 1 to tab1.length do
12:             if tab1[j] < tab1[index] then
13:                 index ← j
14:             end if
15:         end for
16:         smallerNumber ← tab1[index]
17:         tab1[index] ← tab1[i]
18:         TAB[index][0] ← tab1[i]
19:         tab1[i] ← smallerNumber
20:         TAB[i][0] ← tab1[i]
21:     end for
22:     for i ← 0 to tab.length do
23:         for j ← 0 to tab.length do
24:             if TAB[i][0] = tab[j] then
25:                 TAB[i][1] ← j
26:             end if
27:         end for
28:     end for
29:     re ← TAB[indice][1] return re
30: end function
```

Figure III. 11: Algorithme du choix d'offre minimum.

La figure qui va suivre montre le comportement du CSU utilisé lors de l'application de l'algorithme Secure CR :

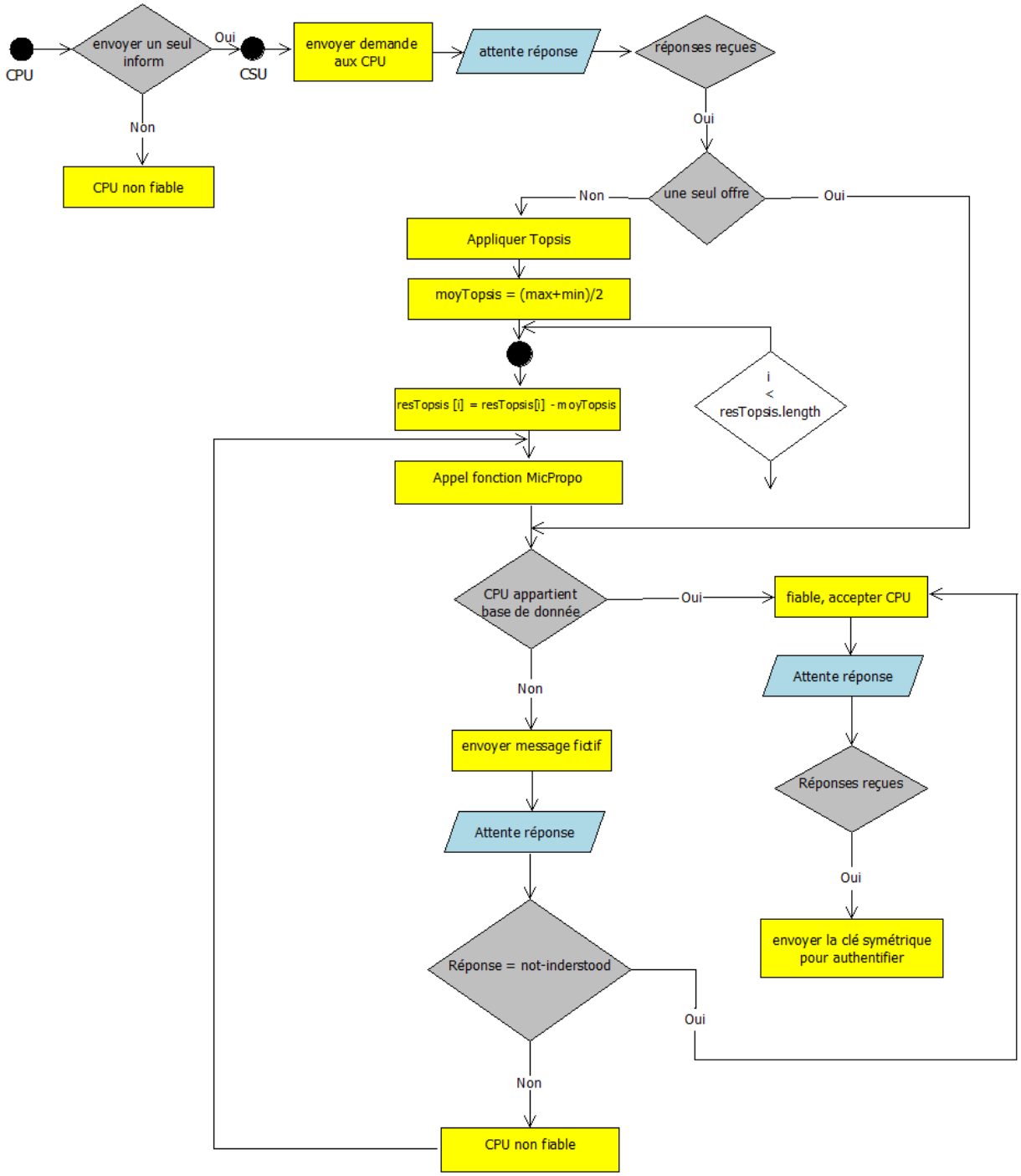

Figure III. 12: Comportement du CSU lors de l'application de l'algorithme Secure CR.

La figure suivante montre le résultat des différentes interactions possibles entre l'utilisateur secondaire, l'utilisateur malveillant et les utilisateurs primaires légitimes lors de l'application de l'algorithme Secure CR.

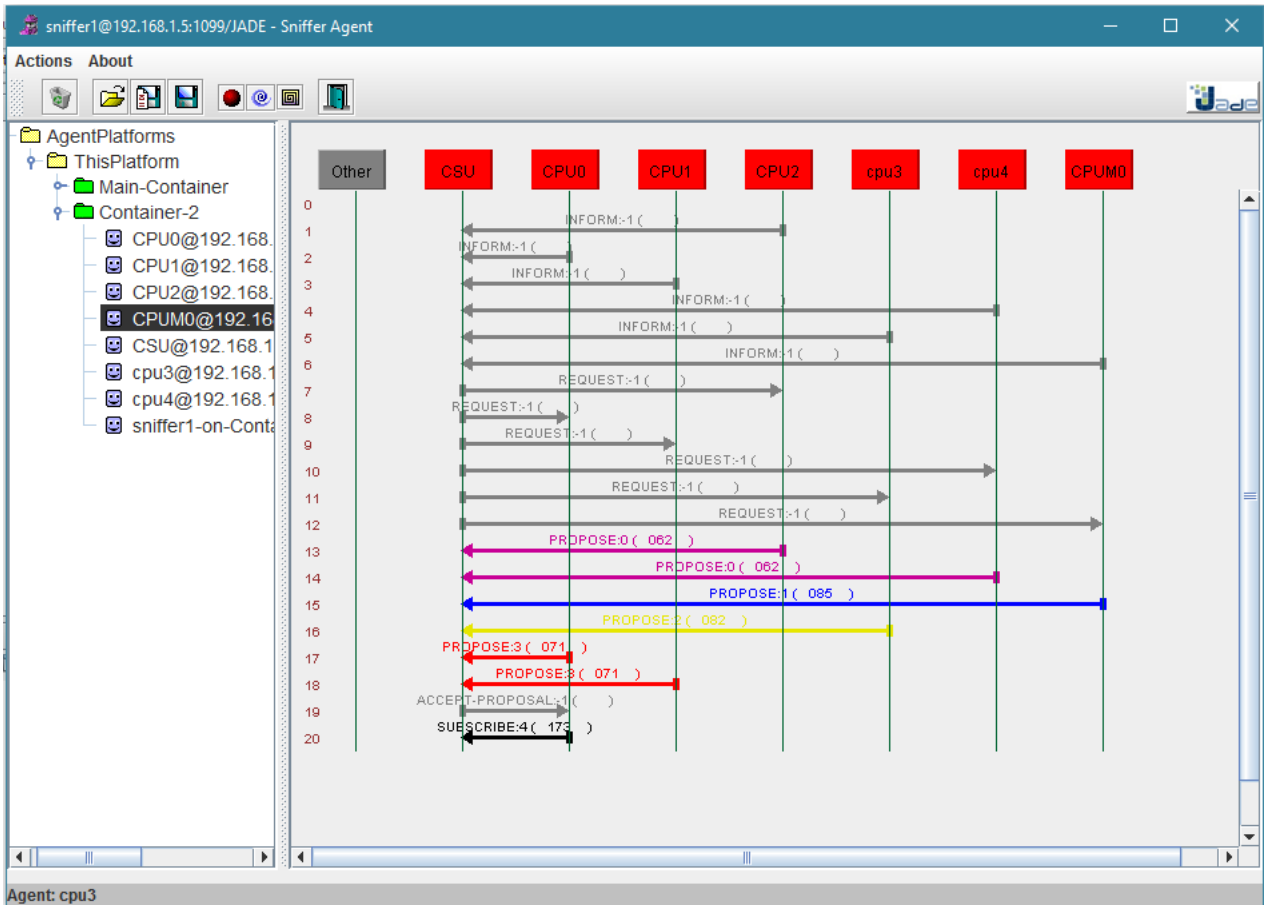

Figure III. 13: Agent Sniffer pour l'algorithme Secure CR.

c. Partie authentification

Pour faire une authentification, nous avons utilisé l'algorithme Blowfish [Annexe B] mais nous l'avons adapté à notre travail. D'abord dans le coté CPU, après que la clé sera générée, elle sera enregistrée en Byte [], ensuite, elle sera convertie et enregistrée en binaire. Cette clé en binaire est la clé de chiffrement et le mot de passe d'authentification en même temps qui sera envoyée dans un message **INFORM**. Ensuite, dans le coté CSU nous avons besoin de l'identificateur du nœud et d'un mot de passe qui sont reçus par le message **INFORM**, après que le CSU ait déterminé le CPU avec qui il va établir une connexion, il lui envoi l'**ACCEPT_PROPOSAL** avec un message qui indique si le CPU choisi est déjà dans la base de données ou non. Si le CPU a déjà été enregistré dans la base de données Agent_CPU, le CPU va répondre avec **SUBSCRIBED** qui a comme paramètre son Id et le mot de passe en Byte [] pour faire une authentification. Sinon le CPU va répondre avec **SUBSCRIBED** qui a comme paramètre son Id et la clé générée en binaire qui sera ajouté dans la base de données du CSU.

III.5.3.Etude comparative

Dans cette section, nous allons élaborer une comparaison en termes de temps d'exécution entre les résultats obtenus avec la méthode Optimal CR, et ceux obtenus par la méthode Secure CR. Une autre comparaison a été réalisée en termes de nombre de messages. Nous allons aussi étudier l'impact du passage à l'échelle dans le cadre de l'utilisation des CPU.

- **Comparaison en termes de temps de traitement**

Afin d'étudier le comportement des deux Algorithmes nous allons fixer les paramètres suivants :

Paramètre	Valeur
Temps d'attente du NOT_UNDERSTOOD	2000 ms
Temps d'attente du PROPOSE	5000 ms

Tableau III. 2: Paramètre de configuration.

Le tableau qui va suivre représente les valeurs obtenues en termes de temps d'exécution pour les deux méthodes :

Nombre de CPU	Temps d'exécution (ms)	
	Méthode Optimal CR	Méthode Secure CR
< 10	5221	5159
10	5442	5249
20	5668	5632
30	5902	5857
40	6148	6137
50	6417	6349
60	6603	6472
70	6838	6710
80	7051	6869
90	7257	7168
100	7356	7280

Tableau III. 3: Meilleur temps obtenue pour les deux méthodes.

La figure ci-dessous présente les courbes de chaque méthode en fonction du temps d'exécution.

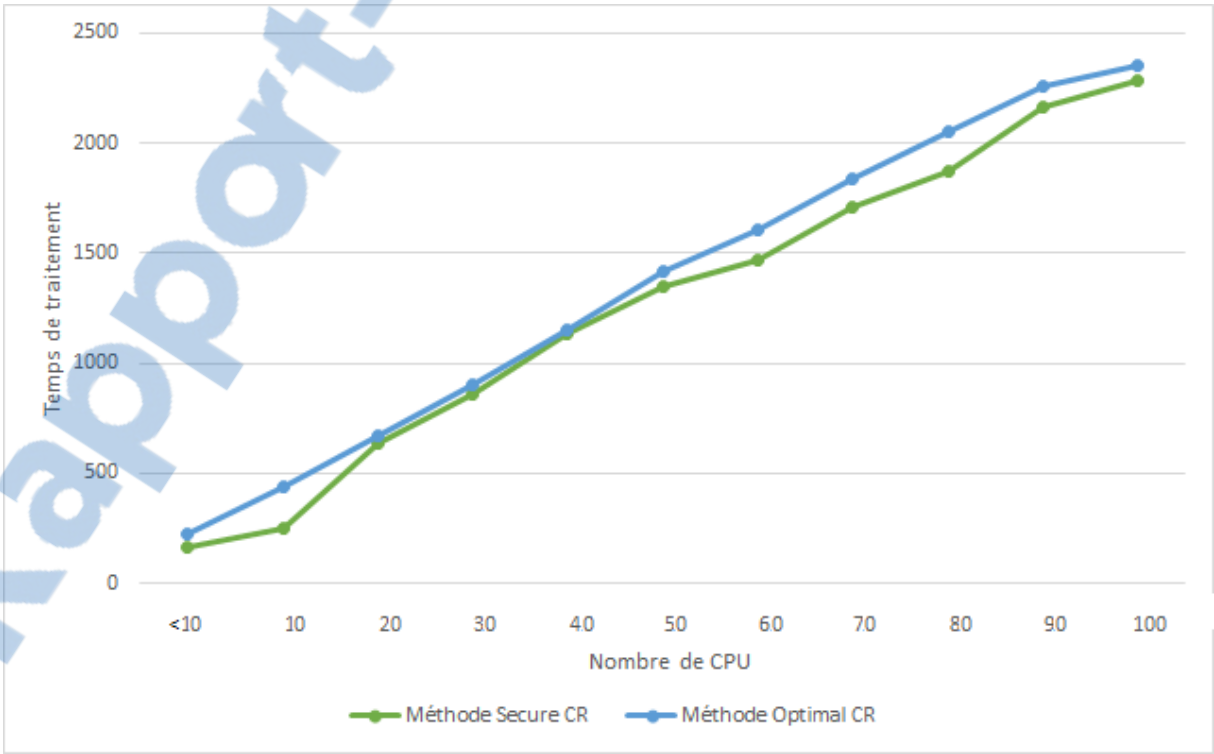

Figure III. 14: Impact du nombre de CPU sur le temps de traitement.

- **Comparaison en termes de nombre de messages**

Le tableau qui va suivre représente le nombre de messages envoyés, reçus et total obtenus par rapport au nombre de CPU pour la méthode Optimal CR.

Méthode Optimal CR	<10	10	20	30	40	50	60	70	80	90	100
Messages Envoyés	7	11	21	31	43	51	61	71	81	91	101
Messages Reçus	7	11	21	31	42	51	58	62	74	80	82
Nombre Messages Total	14	22	42	62	85	102	119	133	155	171	183

Tableau III. 4: Nombre de messages obtenus pour la méthode Optimal CR.

La figure suivante montre les résultats obtenus en termes de nombre de messages.

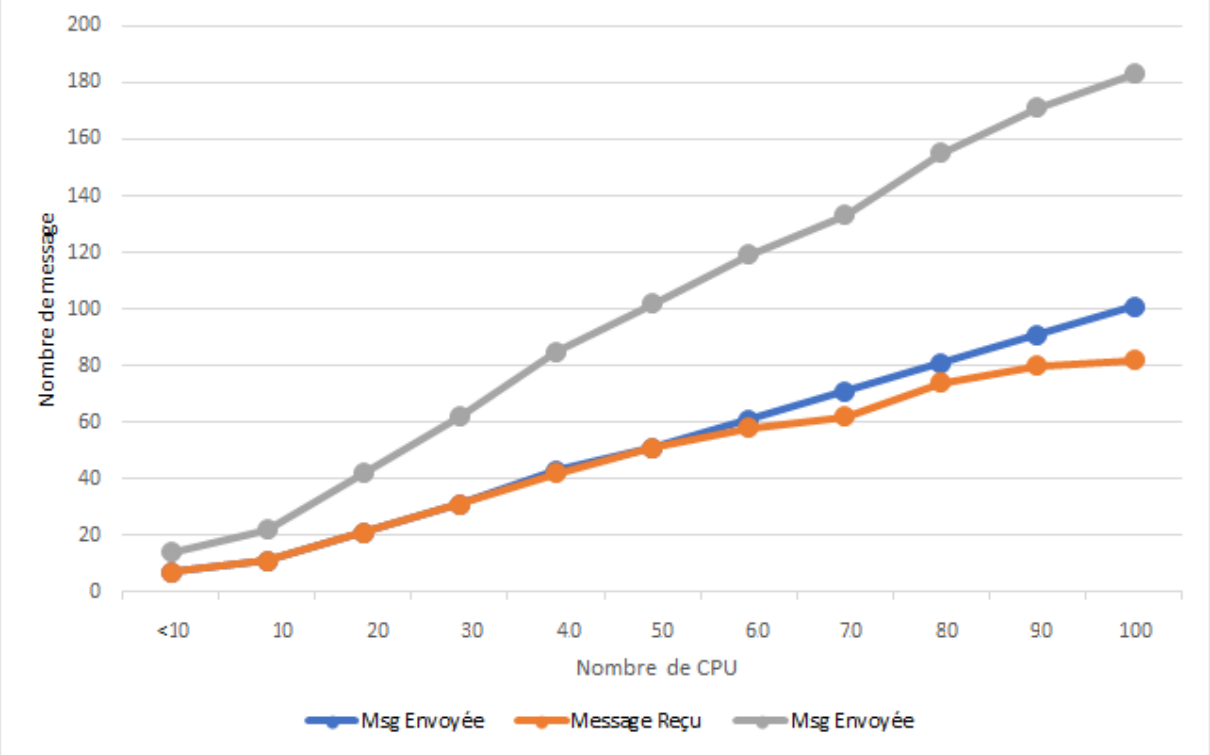

Figure III. 15: Impact du Nombre de CPU sur le Nombre de message dans la méthode Optimal CR.

Le tableau qui va suivre représente le nombre de message envoyé, reçu et total obtenues en termes de temps d'exécution pour la méthode Secure CR :

Méthode Secure CR	<10	10	20	30	40	50	60	70	80	90	100
Message Envoyée	7	11	21	31	41	51	61	71	81	91	102
Message Reçu	7	11	21	31	41	49	56	64	68	75	75
Nombre Message Total	14	22	42	62	82	100	117	135	149	166	177

Tableau III. 5: Nombre de message obtenue pour la méthode Secure CR.

La figure suivante montre les résultats obtenus en termes de nombre de messages.

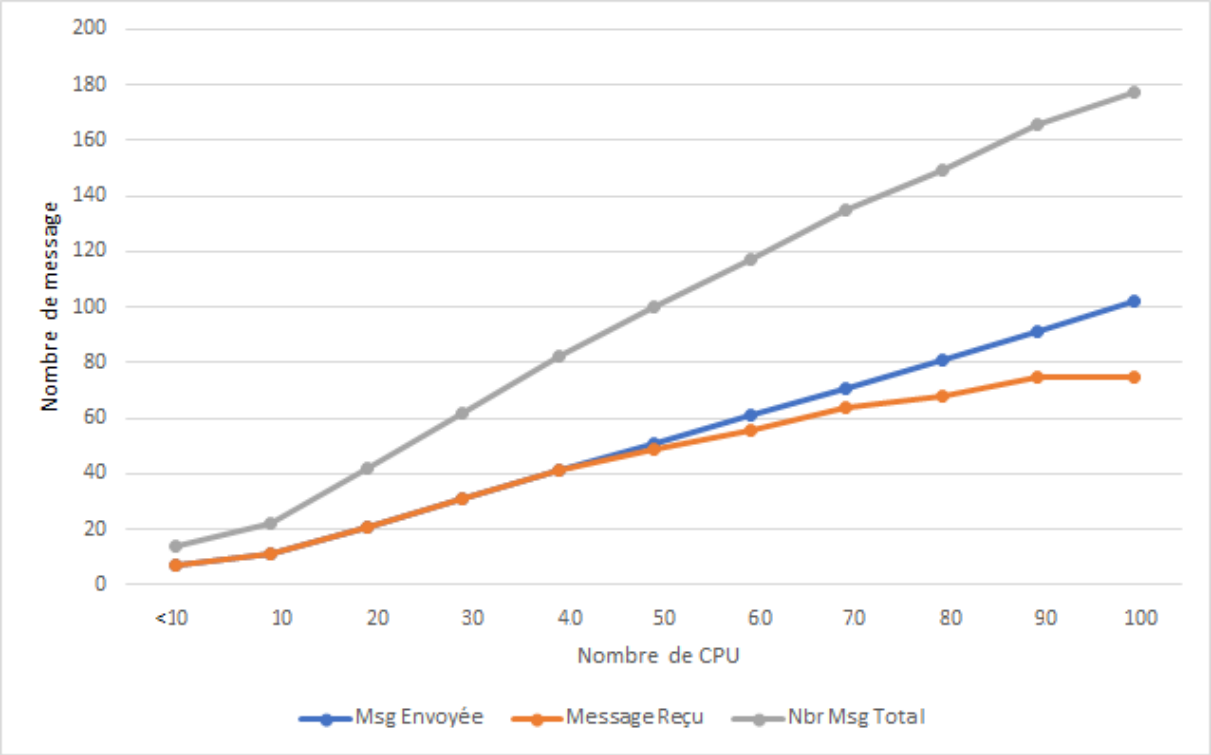

Figure III. 16: Impact du nombre de CPU sur le nombre de message dans la méthode Secure CR.

La figure III.17 montre la différence entre la méthode Optimal RC et Secure RC en termes de nombre de messages.

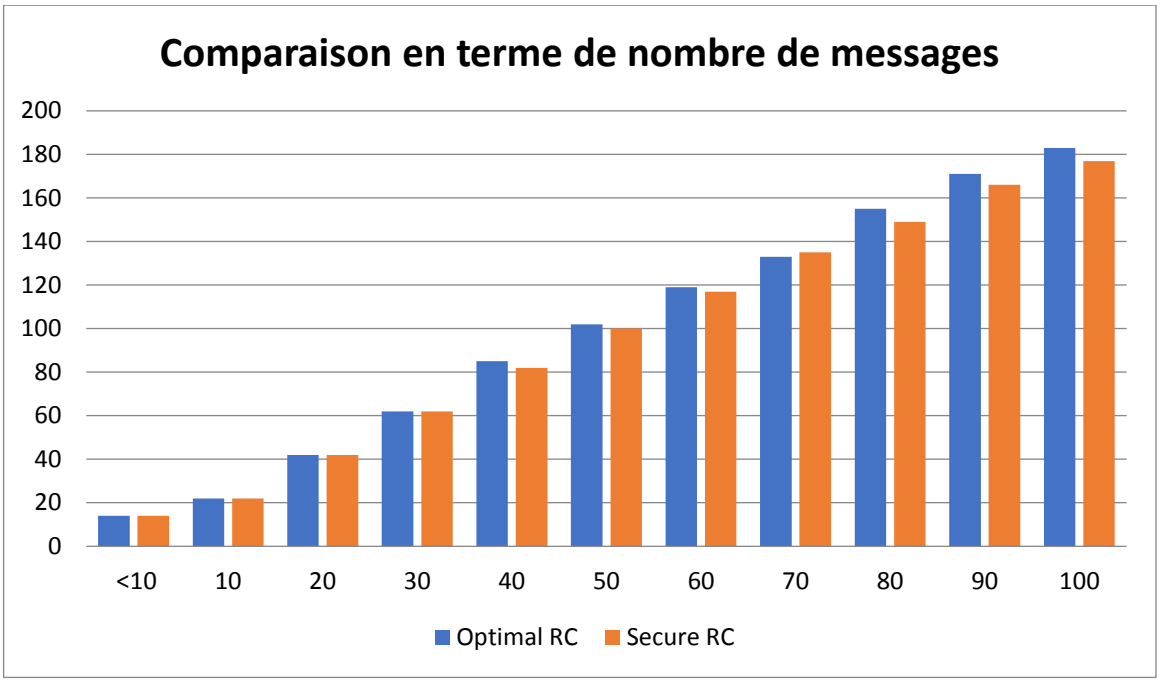

Figure III. 17: Impact du nombre de CPU sur le nombre total des messages.

- **Discussion**

Les résultats obtenus démontrent que l'implémentation des CPU dans l'algorithme Secure CR est plus rapide en termes de temps d'exécution par rapport à l'algorithme Optimal CR. Ainsi que le nombre de message reçu dans la méthode Secure CR est moins que dans la méthode Optimal CR.

Alors, d'après l'implémentation des deux algorithmes et l'étude comparative nous concluons d'une part que l'algorithme Secure CR est plus rapide dans le temps de réponse et plus sécurisé mais il n'est pas optimal car il élimine des offres qui sont meilleures et surtout fiables. D'autre part, l'algorithme Optimal CR est moins rapide dans le temps de réponse, mais plus optimale car il donne la meilleure offre choisie.

Nous remarquons que, à chaque fois que le nombre de CPU augmente, le temps d'exécution augmente et le nombre de messages augmente.

III.5.4.L'interface graphique

Pour bien voir les résultats obtenus, nous avons pensé à les représenter dans une interface graphique claire et simple.

La figure suivante représente l'interface d'accueil qui va nous rediriger vers l'interface de simulation après un simple clic.

Figure III. 18: L'interface d'accueil.

En cliquant sur l'interface d'accueil, l'utilisateur se retrouvera devant la fenêtre de simulation des deux méthodes présentées auparavant.

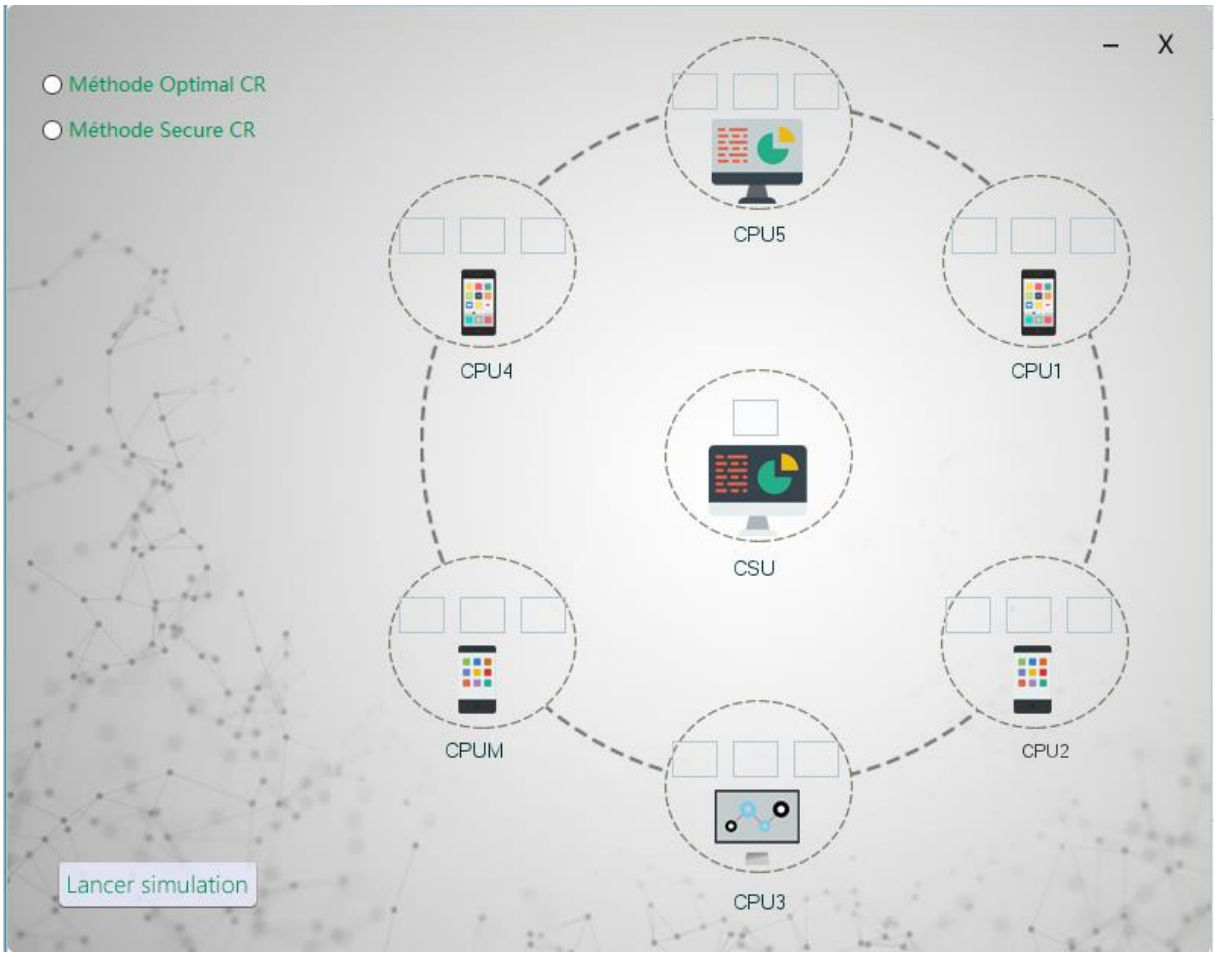

Figure III. 19: L'interface de simulation des deux algorithmes.

Notre interface se compose des utilisateurs primaires légitimes et non-légitimes, un utilisateur malveillant et un utilisateur secondaire. Chaque CPU a trois critères (nombre de canaux, prix, durée) et le CSU a un seul critère à demander (le nombre de canaux dont il a besoin). Aussi deux RadioBoutons pour les deux méthodes : Optimal CR et Secure CR et un bouton pour lancer la simulation.

En choisissant la méthode Optimal CR et en cliquant sur le bouton lancer la simulation, le choix du CPU sera effectuée et apparu en couleur verte et s'il y a un suspect d'être malveillant il sera affiché en rouge, les résultats s'affichent en dessous :

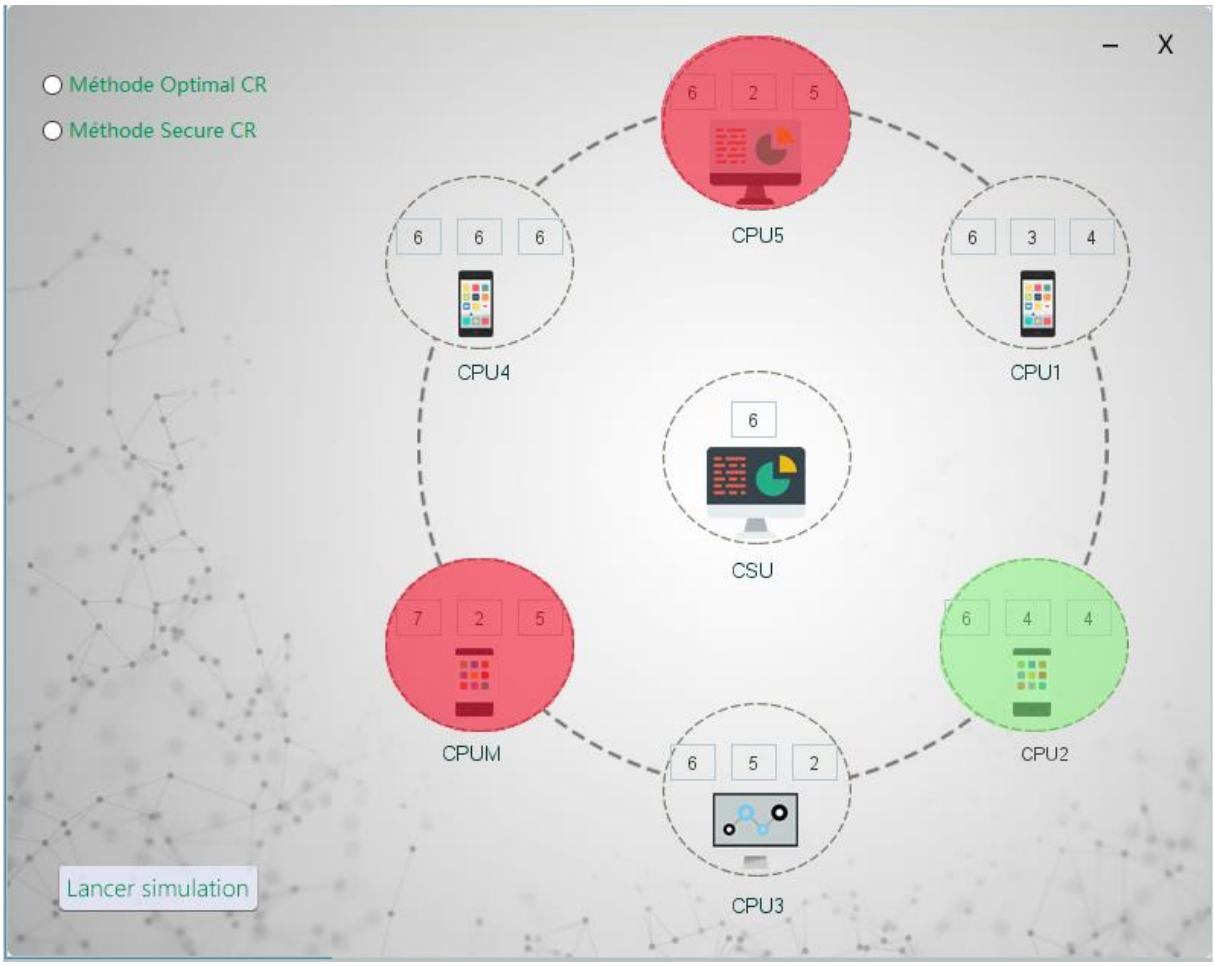

Figure III. 20 : L'exécution de l'algorithme Optimal CR.

III.6. Conclusion

Dans ce chapitre, nous avons présenté les algorithmes utilisés dans notre étude tels que l'algorithme Blowfish qui génère une clé symétrique pour l'authentification et l'algorithme de la décision multicritère TOPSIS pour choisir une meilleure offre. Ensuite, nous avons proposés deux méthodes pour sécuriser la communication dans un réseau radio cognitif.

D'après les résultats obtenus, nous avons remarqué que la méthode Optimal CR donne une meilleure offre donc elle garantit une bonne qualité de service. Par contre, la méthode Secure CR élimine plusieurs offres légitimes mais elle est plus fiable.

Conclusion générale

La radio cognitive est une nouvelle technologie qui permet la détection du spectre et l'utilisation optimale des bandes de fréquence. Cette technologie consiste à partager le spectre entre un utilisateur primaire et un utilisateur secondaire. Le RRC est un aspect de communication sans fil qui permet une détection d'une manière intelligente des canaux de communications. Par conséquent, l'accès au spectre peut être simplement ciblé par différentes attaques.

Puisque le spectre du RRC est ciblé par les attaques du réseau radio cognitif, dans notre étude nous traitons les attaques de type PUE et Hello flood. Donc, il faut des solutions et des approches à suivre pour une utilisation fiable et optimale de cette technologie.

Dans le but de sécuriser l'accès dynamique au spectre dans un réseau radio cognitif, nous avons instauré dans ce mémoire deux algorithmes qui sont : l'algorithme Optimal CR et l'algorithme Secure CR

Ces deux algorithmes permettent d'avoir une liaison fiable entre les utilisateurs primaires et les utilisateurs secondaires, et donc résoudre le problème des attaques du réseau radio cognitif.

Notre étude a prouvé que l'algorithme Optimal CR permet de sécuriser l'accès au spectre d'une façon optimale et il donne un meilleur résultat. Le deuxième qui est Secure CR a le même objectif que le premier algorithme mais d'une manière plus fiable et il ne donne pas une meilleure offre.

Les résultats obtenus montrent que l'algorithme Secure CR est plus performant en termes de temps de réponses par rapport à l'algorithme Optimal CR. Ces résultats approuvent une meilleure qualité de service et une communication fiable dans un réseau radio cognitif.

A la fin et comme perspective nous pouvons utiliser les mêmes algorithmes suggérés dans un scénario qui contient plusieurs CSU. Ainsi, nous pensons à utiliser un apprentissage réel avec des données réelles. Aussi, nous pouvons traiter les autres types d'attaques qui ont été citées dans le chapitre 2 et de penser de travailler sur le cas des nœuds mobiles.

BIBLIOGRAPHIE

[1] A. Ben Dhaou, « Allocation dynamique des bandes spectrales dans les réseaux sans-fil à radio cognitive », 2011. [En ligne]. Disponible sur : http://www.archipel.uqam.ca/id/eprint/4368. [Consulté le : 02-février-2017].

[2] J. Mitola et G. Q. Maguire, « Cognitive radio : making software radios more personal », *IEEE Pers. Commun.*, vol. 6, n° 4, p. 13–18, 1999.

[3] A. Amraoui, W. Baghli, et B. Benmammar, « Amélioration de la fiabilité du lien sans fil pour un terminal radio cognitive mobile », in *Les 12èmes Journées Doctorales en Informatique et Réseaux (JDIR'11)*, 2011.

[4] B. Benmammar, « Présentation de la radio cognitive », p. 28, 2012. [En ligne]. Disponible sur : https://hal.archives-ouvertes.fr/docs/00/68/23/44/PDF/RC-Pres-Benmammar.pdf. [Consulté le : 03-février-2017].

[5] D. GRANDBLAISE, « Partage distribué et dynamique de spectre entre cellules par l'utilisation de jetons crédits », *Com. Natl. Fr. Radioélectr. Sci.*, 2006.

[6] J. O. Neel, « Analysis and design of cognitive radio networks and distributed radio resource management algorithms », Virginia Polytechnic Institute and State University, 2006.

[7] A. Amraoui, B. Benmammar, et F. T. Bendimerad, « Accès Dynamique au Spectre dans le Contexte de la Radio Cognitive », in *2ième édition de la conférence nationale de l'informatique destinée aux étudiants de graduation et de post-graduation*, 2012.

[8] E. Trigui, M. Esseghir, et L. M. Boulahia, « Gestion dynamique du spectre entre terminaux radio cognitive mobiles », in *CFIP 2011-Colloque Francophone sur l Ingénierie des Protocoles*, 2011.

[9] W. El-Hajj, H. Safa, et M. Guizani, « Survey of security issues in cognitive radio networks », vol. 12, n° 2, p. 181–198, 2011.

[10] S. Gambs, « Authentification de messages et mots de passe - Recherche Google », 09-nov-2015. [En ligne]. Disponible sur: https://www.google.com/search?q=Authentification+de+messages+et+mots+de+passe&ie=utf-8&oe=utf-8&client=firefox-b. [Consulté le: 11-mars-2017].

[11] C. N. Mathur et K. P. Subbalakshmi, « Security issues in cognitive radio networks », *Cogn. Netw. Self-Aware Netw.*, p. 284–293, 2007.

[12] C. Braz, « AuthenLink : Un système d'authentification centré-utilisateur pour un commerce mobile sécurisé ». [En ligne]. Disponible sur : https://www.researchgate.net/profile/Christina_Braz/publication/267691458_AuthenLink_Un_systeme_d'authentification_centre_utilisateur_pour_un_commerce_mobile_securise/links/5686cc7808aebccc4e13c81b.pdf. [Consulté le : 25-février-2017].

[13] R. Chen et J.-M. Park, « Ensuring trustworthy spectrum sensing in cognitive radio networks », in *Networking Technologies for Software Defined Radio Networks, 2006. SDR'06.1 st IEEE Workshop on*, 2006, p. 110–119.

[14] Y. Zhang, G. Xu, et X. Geng, « Security threats in cognitive radio networks », in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, 2008, p. 1036–1041.

[15] R. Chen, « Enhancing attack resilience in cognitive radio networks », Virginia Tech, 2008. [En ligne]. Disponible sur : https://vtechworks.lib.vt.edu/handle/10919/26330. [Consulté le : 05-mars-2017].

[16] T. C. Clancy et N. Goergen, « Security in cognitive radio networks: Threats and mitigation », in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, 2008, p. 1–8.

[17] O. León, J. Hernández-Serrano, et M. Soriano, « Securing cognitive radio networks », *Int. J. Commun. Syst.*, vol. 23, n° 5, p. 633–652, 2010.

[18] R. Chen, J.-M. Park, Y. T. Hou, et J. H. Reed, « Toward secure distributed spectrum sensing in cognitive radio networks », *IEEE Commun. Mag.*, vol. 46, n° 4, 2008.

[19] R. Chen, J.-M. Park, et J. H. Reed, « Defense against primary user emulation attacks in cognitive radio networks », *IEEE J. Sel. Areas Commun.*, vol. 26, n° 1, 2008.

[20] O. R. Afolabi, K. Kim, et A. Ahmad, « On secure spectrum sensing in cognitive radio networks using emitters electromagnetic signature », in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, 2009, p. 1–5.

[21] O. Ureten et N. Serinken, « Wireless security through RF fingerprinting », *Can. J. Electr. Comput. Eng.*, vol. 32, n° 1, p. 27–33, 2007.

[22] C. Zhao, W. Wang, L. Huang, et Y. Yao, « Anti-PUE attack base on the transmitter fingerprint identification in cognitive radio », in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, 2009, p. 1–5.

[23] Q. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*. John Wiley & Sons, 2007.

[24] W. Xu, W. Trappe, Y. Zhang, et T. Wood, « The feasibility of launching and detecting jamming attacks in wireless networks », in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, p. 46–57.

[25] W. Xu, T. Wood, W. Trappe, et Y. Zhang, « Channel surfing and spatial retreats: defenses against wireless denial of service », in *Proceedings of the 3rd ACM workshop on Wireless security*, 2004, p. 80–89.

[26] O. Leon, J. Hernandez-Serrano, et M. Soriano, « A new cross-layer attack to TCP in cognitive radio networks », in *Cross Layer Design, 2009. IWCLD'09. Second International Workshop on*, 2009, p. 1–5.

[27] C. Karlof et D. Wagner, « Secure routing in wireless sensor networks: Attacks and countermeasures », *Ad Hoc Netw.*, vol. 1, n° 2, p. 293–315, 2003.

[28] P. Anand, A. S. Rawat, H. Chen, et P. K. Varshney, « Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks », in *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, 2010, p. 1–9.

[29] H. Wang, L. Lightfoot, et T. Li, « On phy-layer security of cognitive radio: Collaborative sensing under malicious attacks », in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, 2010, p. 1–6.

[30] Y. Shei et Y. T. Su, « A sequential test based cooperative spectrum sensing scheme for cognitive radios », in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, 2008, p. 1–5.

[31] A. S. Rawat, P. Anand, H. Chen, et P. K. Varshney, « Countering byzantine attacks in cognitive radio networks », in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, p. 3098–3101.

[32] W. Wang, H. Li, Y. Sun, et Z. Han, « Attack-proof collaborative spectrum sensing in cognitive radio networks », in *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, 2009, p. 130–134.

[33] K. Bian et J.-M. Park, « MAC-layer misbehaviors in multi-hop cognitive radio networks », in *2006 US-Korea Conference on Science, Technology, and Entrepreneurship (UKC2006)*, 2006.

[34] J. Hernandez-Serrano, O. León, et M. Soriano, « Modeling the lion attack in cognitive radio networks », *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, p. 2, 2011.

[35] G. A. Safdar, S. Albermany, N. Aslam, A. Mansour, et G. Epiphaniou, « Prevention against threats to self co-existence-A novel authentication protocol for cognitive radio networks », in *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, 2014, p. 1–6.

[36] S. Chandrashekar, « PRIMARY USER AUTHENTICATION METHODS FOR MOBILE COGNITIVE RADIO NETWORKS », UNIVERSITY OF ARIZONA, 2012.

[37] J.-C. Chen et Y.-P. Wang, « Extensible authentication protocol (EAP) and IEEE 802.1 x: tutorial and empirical experience », *IEEE Commun. Mag.*, vol. 43, n° 12, p. supl–26, 2005.

[38] « Media Security and Reliability Council », *Commun. Infrastructurevsecurity Access Restor. Work. Group final Rep.*

[39] A. AMRAOUI, « Vers une architecture multi-agents pour la radio cognitive opportuniste », Thèse de Doctorat, Université de Tlemcen, 2015.

[40] F. Jacques, « Les Systèmes Multi-agents, Vers une intelligence collective », *InterEditions Paris*, vol. 322, 1995.

[41] F. L. Bellifemine, G. Caire, et D. Greenwood, *Developing multi-agent systems with JADE*, vol. 7. John Wiley & Sons, 2007.

[42] O. Boissier, I. Vue d'ensemble, I. I. Caractéristiques, et I. P. JADE, « Systèmes multi-agents », 2010.

[43] tutorialspoint.com, « SQLite Overview », *www.tutorialspoint.com*. [En ligne]. Disponible sur: https://www.tutorialspoint.com/sqlite/sqlite_overview.htm. [Consulté le: 02-mai-2017].

[44] « NetBeans IDE - Java EE Development ». [En ligne]. Disponible sur: https://netbeans.org/features/java-on-server/java-ee.html. [Consulté le: 02-mai-2017].

[45] « Schneier on Security: The Blowfish Encryption Algorithm ». [En ligne]. Disponible sur: https://www.schneier.com/academic/blowfish/. [Consulté le: 02-juin-2017].

[46] « Academic: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) - Schneier on Security ». [En ligne]. Disponible sur: https://www.schneier.com/academic/archives/1994/09/description_of_a_new.html. [Consulté le: 02-juin-2017].

Annexe A

L’algorithme TOPSIS

La méthode « TOPSIS » a été développée à l’origine par Hwang et Yoon en 1981 avec, notamment, d’autres développements de Yoon en 1987, et Hwang, Lai et Liu en 1993.

TOPSIS est une méthode d’analyse multicritères pour l’aide à la prise de décision. L’idée principale de cette méthode est de choisir l’action ayant :

La plus petite distance à l’action dite « idéale » (positive-ideal solution).

La plus grande distance à l’action dite « anti-idéale » (negative-ideal solution).

La figure suivante décrit les différentes étapes de l’algorithme TOPSIS :

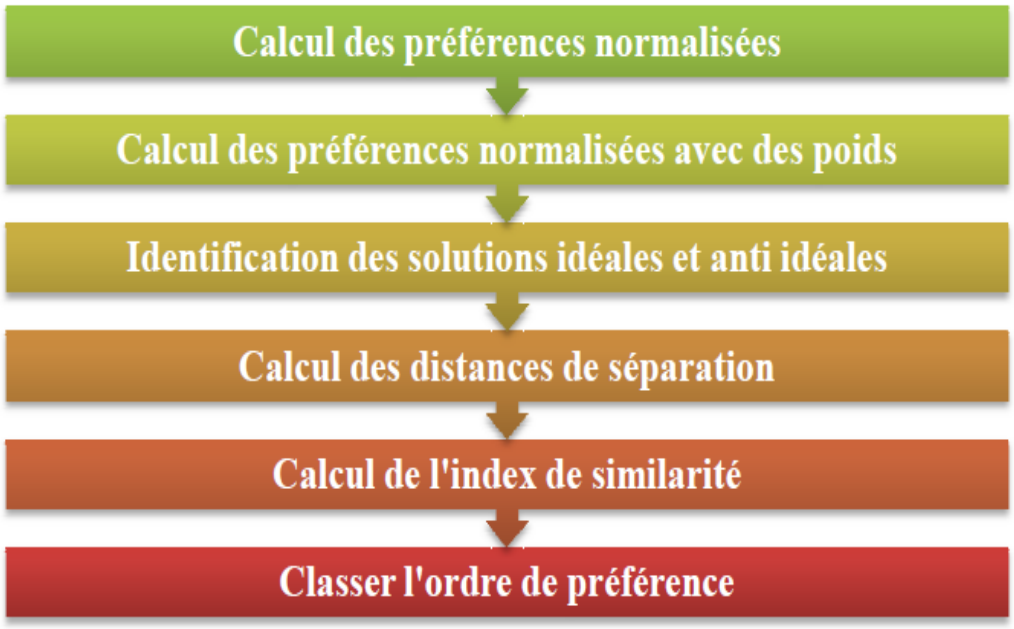

Figure A 1: Différentes étapes de l’algorithme TOPSIS.

Dans ce qui suit, nous allons détailler les différentes étapes de l’algorithme TOPSIS, un exemple applicatif est donné en annexe.

Étape 1 : Calcul des préférences normalisées

La construction d'une matrice normalisée pour transformer l'attribut à diverses dimensions en attributs adimensionnels, ce qui permet une comparaison entre les attributs.

$$\boldsymbol{r_j(x_i)} = \frac{\boldsymbol{g_j(x_i)}}{\sqrt{\boldsymbol{\Sigma_{i=1}^{m}\left(g_j(x_i)\right)^2}}} \quad \text{i} = 1..\text{m}, \text{j} = 1..\text{n}$$

Étape 2 : Calcul des préférences normalisées avec des poids associés aux critères

La construction d'une matrice normalisée et pondérée.

$$V = \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mn} \end{bmatrix} = \begin{bmatrix} W_1.r_{11} & \cdots & W_n.r_{1n} \\ \vdots & \ddots & \vdots \\ W_1.r_{m1} & \cdots & W_n.r_{mn} \end{bmatrix}$$

$V_j(x_i) = w_j\, r_j(x_i) \qquad i = 1, \ldots, m.\, j = 1, \ldots, n$

$w_j = \{ w_1, w_2, \ldots . w_n \}$: L'ensemble des poids associés aux critères.

Étape 3 : Identification des solutions idéales et anti-idéales

$$A^* = \left\{v_1^*, \ldots, v_j^*, \ldots, v_n^*\right\} = \{(\text{Max}_i\ v_j(x_i)\ /j \epsilon J_1), (Min_i\ v_j(x_i)/\ \ j\ \epsilon J_2)\}$$

$$A' = \left\{v_1', \ldots, v_j', \ldots, v_n'\right\} = \{(Min_i v_j(x_i)\ /\ j \epsilon J_1), (Max_i v_j(x_i)\ /j \epsilon J_2)\}$$

J_1 : ensemble des critères de bénéfice J_2 : ensemble des critères de coût

Étape 4 : Calcul des distances de séparation

- Distance idéale

$$d^*(x_i) = \sqrt{\sum_{j=1}^{n}\left(v_j^* - v_{ij}\right)^2}$$

- Distance anti-idéale

$$d'(x_i) = \sqrt{\sum_{j=1}^{n}\left(v' - v_{ij}\right)^2}$$

Étape 5 : Calcul de l'indice de similarité à la solution idéale

$$c(x_i) = \frac{d'(x_i)}{(d^*(x_i) + d'(x_i))} \; 0 < c(x_i) < 1 \quad i = 1..m$$

$c(x_i) = 1 \; Si \; A(x_i) = A^* \; (solution \; idéale)$

$c(x_i) = 1 \; Si \; A(x_i) = A' \; (solution \; anti - idéale)$

Étape 6 : Classer l'ordre de préférence

- Choisir l'action ayant le plus grand index de similarité.

- Ranger les actions par ordre décroissant des index de similarité.[39]

Annexe B

L’algorithme Blowfish

Blowfish est un chiffrement de bloc symétrique qui peut être utilisé pour le cryptage et la sauvegarde des données. Il faut une clé à longueur variable, de 32 bits à 448 bits, ce qui le rend idéal pour une utilisation domestique et exportable. Blowfish a été conçu en 1993 par Bruce Schneier comme une alternative rapide et gratuite aux algorithmes de chiffrement existants. Depuis lors, il a été considérablement analysé, et il accepte lentement l'acceptation comme un algorithme de cryptage fort. Blowfish n'est pas breveté et sans licence, et est disponible gratuitement pour toutes les utilisations.[45]

Blowfish est une clé de longueur variable de 64 bits. L'algorithme se compose de deux parties : une partie d'extension de clé et une partie de chiffrement de données. L'expansion de clé convertit une clé d'au plus 448 bits en plusieurs tableaux de sous-clés totalisant 4168 octets.

Le cryptage des données se fait à 16 ronds. Chaque tour se compose d'une permutation dépendante de la clé et d'une substitution dépendante des clés et des données. Toutes les opérations sont des XOR et des ajouts sur des mots de 32 bits. Les seules opérations supplémentaires sont quatre recherches indexées de données de tableau par tour.

Sous-clés

Blowfish utilise un grand nombre de sous-clés. Ces clés doivent être préalablement calculées avant tout chiffrement ou décryptage de données.

1. Le champ P se compose de 18 sous-clés de 32 bits : P1, P2, ..., P18.
2. Il existe quatre S-Boxes de 256 éléments chacune.

S1,0, S1, 1..., S1,255;
S2,0, S2, 1..., S2,255;
S3,0, S3, 1..., S3,255;
S4,0, S4, 1..., S4,255.

La figure qui suit représente le comportement de l’algorithme Blowfish.

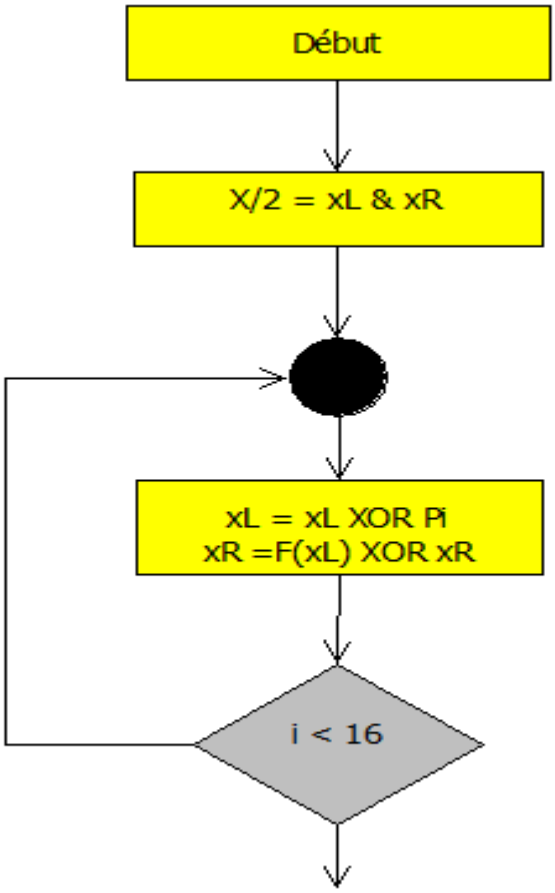

Figure B 1: Comportement de l'algorithme Blowfish.

La fonction F

La F-fonction de Blowfish divise une entrée xL de 32 bits en quatre morceaux de 8 bits : a, b, c et d

$$F(xL) = \left((S1, a + S2, b \bmod 2^{32})\ XOR\ S3, c\right) + S4, d \bmod 2^{32}$$

La représentation de la fonction Blowfish selon la figure suivante :[46]

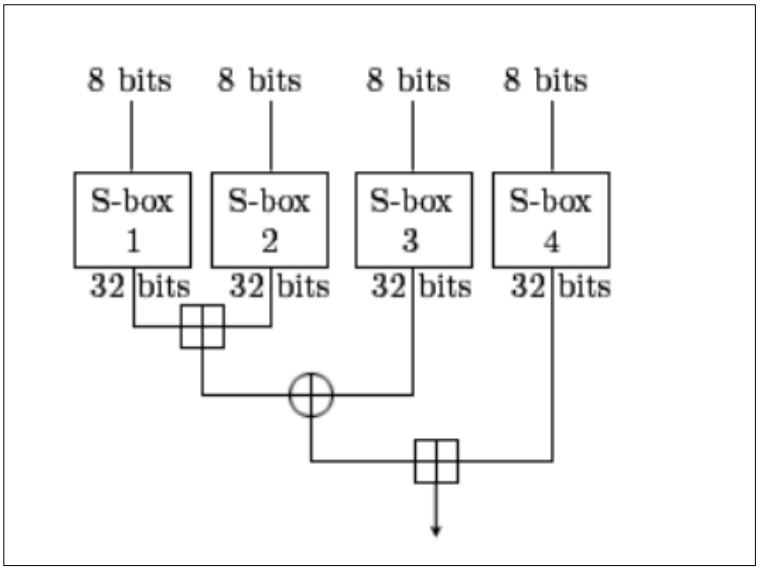

Figure B 2: La fonction Blowfish.