MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Réseaux et Systèmes**

Présenté par :

DIAKITE Madibaba

THEME :

Analyse et contre-mesure des attaques de topologie dans les réseaux LLN

Soutenu le : 19/06/2021

Devant le jury composé de :

Mme Fillali Fatima Zohra	MCB	Université de Mostaganem	Présidente
Mme Benidris Fatima Zohra	MCB	Université de Mostaganem	Examinatrice
Mme Abid Meriem	MCB	Université de Mostaganem	Encadrante

Année Universitaire 2020-2021

Dédicaces

Je dédie ce modeste travail :

À ma mère

À mon père

À mes oncles Mangassouba Hamet et Gary Bakary

À mon cousin Doucouré Papa

À mon grand-père Wagué Lagami

je pourrais ne jamais exprimer le respect que j'ai pour vous. Vos prières, vos encouragements et votre soutien m'ont toujours été d'un grand secours. Puisse Dieu, le tout puissant vous préserver du mal, vous combler de santé, de bonheur et vous procurer une longue vie.

Remerciements

Dieu merci pour la santé, la volonté, le courage et la détermination qui m'ont permis d'achever ce modeste travail.

J'adresse toute ma reconnaissance et gratitude à mon encadrante ***M****adame Abid Meriem*, pour sa patience, sa disponibilité, ses orientations, et surtout ses judicieux conseils, qui ont contribué à réaliser ce projet.

Je remercie également mes parents, pour leur soutien constant et leurs encouragements et tout les autres membres de ma famille.

Enfin, je tiens à remercier tout le personnel administratif de la faculté des sciences exactes et de l'informatique.

Resumé

L'Internet des objets (IoT), grâce à sa large gamme d'applications, est l'une des technologies les plus prometteuses de notre époque. Les réseaux à faible puissance et à fort taux de perte (LLN) forment une partie de l'IoT et sont des capteurs et des actionneurs aux ressources limitées reliés par des liaisons instables. Le protocole de routage RPL a été développé pour s'adapter aux fortes contraintes de ces réseaux. Cependant, RPL est exposé à une grande variété d'attaques. Dans ce mémoire, nous nous intéressons aux attaques de topologie, plus précisement aux attaques de diminution et d'augmentantion de rang, dans des LLN ainsi que leurs techniques de prévention, de détection et d'atténuation dans la littérature. Puis nous proposerons et évaluerons notre propre solution pour contrer ces attaques.

Mots-clés : Internet des Objets, LLN, RPL, Sécurité, attaques, contre-mesures, simulation, évaluation

Liste des figures

Liste des tableaux

Liste des abréviations

DODAG Destination Oriented Directed Acyclic Graph

LLN Low Power and Lossy Network

DIO DODAG Information Object

DIS DODAG Information Solicitation

DAO DODAG Advertisement Object

MP2P Multipoint-to-point

P2MP Point-to-multipoint

P2P Point-to-point

IoT Internet of Things

OF0 Objective Function Zero

MRHOF Minimum Rank with Hysteresis Objective Function

ROLL Routing Over Low power and Lossy networks

IETF Internet Engineering Task Force

IPv6 Internet Protocol version 6

ROLL Routing Over Low power and Lossy networks

VeRA Version Number and Rank Authentication

TRAIL TRustANCHORInterconexionLoop

RFC Requests For Comments

ETX Expected Transmission Count

TP Vrai Positif

TN Vrai Négatif

FP Faux Positif

FN Faux Négatif

TFN Taux Faux Négatif

TFP Taux Faux Positif

TTP Taux Vrai Négatif

FAR Taux Fausse Alerte

Table des matières

Introduction générale

L'émergence de l'Internet des objets (IoT) et de ses nombreuses applications prometteuses a retenu l'attention de plusieurs chercheurs, organismes de normalisation et universités.

Les réseaux LLN (Low Power and Lossy Networks ou réseaux à faible puissance et à fort taux de perte) forment une partie l'IoT et sont fortement contraints en terme de puissance de traitement, d'énergie, et de mémoire. Ils sont caractérisés par des liaisons instables avec des taux de perte élevés. Le protocole de routage RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) a été spécialement conçu pour prendre en charge les exigences spécifiques de ces réseaux. Les réseaux LLN sont exposés à une grande variété d'attaques à cause de leurs ressources limitées.

L'objectif principal de notre travail est d'étudier et d'analyser dans un premier temps les attaques de topologie dont fait l'objet le protocole RPL ainsi que les solutions proposées dans la littérature pour y faire face. Ensuite, nous proposerons une solution pour mitiger ces attaques, puis nous l'implémenterons et l'évaluerons à l'aide du simulateur COOJA.

Notre mémoire est divisé en 4 chapitres et structuré comme suit :

Dans le premier chapitre, nous nous intéresserons aux réseaux LLN en présentant leurs caractéristiques, leurs contraintes et leurs multiples applications. Ensuite nous étudierons le fonctionnement de son protocole de routage.

Dans le second chapitre, nous nous focaliserons sur la sécurité des réseaux LLN. Puis nous aborderons les attaques qui menacent ces réseaux ainsi que les mécanismes utilisés pour les contrer.

Nous proposerons notre solution dans le troisième chapitre.

Dans le dernier chapitre, nous réaliserons une étude expérimentale. Nous simulerons des LLN afin d'évaluer les performances de rpl et l'impact des attaques contre rpl puis nous évaluerons notre solution.

Ainsi, nous terminerons notre mémoire par une conclusion générale et quelques perspectives.

Chapitre I

Les réseaux LLN

1. Introduction

Dans ce chapitre, nous nous situons dans le contexte des réseaux LLN en présentant leurs domaines d'application, leurs exigences. Puis nous abordons le routage dans ces réseaux. Nous détaillons le fonctionnement du protocole de routage dédié à ces derniers en parlant des mécanismes de sécurité, des modes d'opération, et des modèles de communication spécifiés par ce protocole. Nous conclurons en dernier lieu.

2. L'Internet des objets

L'Internet des objets (IoT) est un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant [1]. Il est un réseau d'appareils dans lequel des objets intelligents sont inter-connectés et permettent de collecter et d'échanger des informations via Internet. Il pourrait être décrit comme le réseau omniprésent et mondial qui aide et fournit un système pour la surveillance et le contrôle du monde physique grâce à la collecte, au traitement et à l'analyse des données générées par des capteurs IoT.

3. Domaines d'application

Les applications potentielles de l'IoT sont nombreuses et diverses, pénétrant pratiquement tous les domaines de la vie quotidienne des individus, des entreprises et de la société dans son ensemble. Voici quelques-unes des applications IoT.

- Smart Home (maison intelligente) : nous pouvons contrôler les appareils (lampes, réfrigérateur, chauffage etc.), surveiller le linge à distance pour éviter les accidents et économiser de l'énergie, ouvrir les fenêtres et les portes etc.
- Smart City (ville intelligente) : nous pouvons surveiller en temps réel la disponibilité des places des parkings, niveaux de déchets dans les conteneurs pour optimiser les itinéraires de collecte des déchets.
- Smart health : les objets connectés faciliteront le suivi des conditions des patients à l'intérieur des hôpitaux et des maisons de retraite, le contrôle des conditions des vaccins et des médicaments à l'intérieur des congélateurs etc.
- Smart agriculture : les objets connectés nous permettront d'obtenir des données précises sur l'état du sol d'un champs (pour plus tard maximiser la production de fruits et légumes), de localiser des animaux dans les pâturages etc.

4. IPv6 et les IoT

L'IoT comprend un nombre incroyablement élevé de nœuds, chacun des nœuds doit être identifiable et accessible par tout utilisateur autorisé, quelle que soit sa position. Pour remédier à cela, l'adressage IPv6 (Internet Protocol version 6) a été proposé pour l'IoT. Les adresses IPv6 sont exprimées au moyen de 128 bits et ce qui devrait suffire à identifier tout objet qui mérite d'être adressé.

5. Les réseaux LLN

Les réseaux LLN (Low Power and Lossy Networks ou réseaux à faible puissance et à fort taux de perte) forment une partie de l'internet des objets. Ce sont des réseaux dans lesquels les nœuds et leurs liaisons sont fortement contraints. Les nœuds fonctionnent avec de forte

contraintes sur la puissance de traitement, l'énergie, et la mémoire. Quant aux liaisons, elles sont instables et sont caractérisées par des taux de perte élevés et des débits faibles [3]. Plusieurs types de liaison sont utilisés dans ces réseaux, notamment IEEE 802.15.4, 6LOWPAN, Zigbee, Bluetooth à basse consommation, Wifi low power etc.

6. Routage dans les LLN

Les LLN ont récemment attiré l'attention des universités et des organismes de normalisation, dans le but de développer des solutions de routage qui garantissent une utilisation efficace des ressources réseau limitées.

L'IETF a effectué des tests sur les protocoles de routage existants spécifiés dans les RFC afin de savoir si l'un d'eux répond aux exigences des LLN. Les résultats ont montré qu'aucun protocole ne répondait spécifiquement aux contraintes de routage dans les LLN (contrainte d'énergie, de lien, de calcul et de mémoire)[22] .

C'est pourquoi le protocole de routage RPL a été spécialement conçu pour répondre aux fortes contraintes de ces réseaux. Il est proactif et basé sur un algorithme à vecteur de distance. Il a été standardisé par le groupe ROLL (Routing Over Low power and Lossy networks) de l'IETF (Internet Engineering Task Force) sous la référence *RFC6550*.

Un protocole est dit proactif lorsque chaque nœud connait les routes vers n'importe quelle destination du réseau à tout instant. Les nœuds calculent les routes avant d'avoir besoin de ces dernières et mettent à jour régulièrement leur table de routage.

Les protocoles à vecteur de distances permettent de créer des tables de routages où aucun nœud n'a la cartographie complète du réseau. Cela permet aux nœuds de fonctionner avec un minimum de ressources. Dans la section ci-après, nous détaillons le fonctionnement de RPL.

7. Protocole de routage de RPL

Nous expliquons dans cette partie les points essentiels du fonctionnement de RPL.

7.1. Construction de la topologie

RPL s'appuie sur la construction d'une structure logique appelée DODAG (Destination Oriented Directed Acyclic Graph). Un DODAG est un graphe acyclique et orienté vers une destination. Dans chacun de ces DODAG, un nœud est considéré comme racine (DODAG root). La racine diffuse un message DIO qui contient des informations sur le DODAG comme le numéro de version, l'identifiant de l'instance et la fonction objective utilisée. Une instance RPL est un ensemble de DODAG partageant une même fonction objective et un identifiant appelé RPLInstanceID.

Lorsque le DIO est reçu par un nœud, ce dernier calcule son rang en fonction de la fonction objective de sorte que son rang soit supérieur à celui de son parent. Ensuite il ajoute l'expéditeur du message à sa liste de parents. Le récepteur ignore le DIO dans le cas où son rang est inférieur à celui de l'expéditeur. Nous reviendrons sur les messages de contrôle dans la sous-section 7.3.

Quand un nouveau nœud désire joindre un DODAG déjà formé, il diffuse un message DIS pour obtenir les informations de configuration afin de se rattacher au DODAG. Il recevra alors des messages DIO venant de plusieurs nœuds en réponse.

Pour maintenir le DODAG (configurer les routes et annoncer les changements), les DIO sont envoyés périodiquement. Trickle est l'algorithme utilisé pour optimiser la fréquence de transmission des DIO en fonction de l'état du réseau[17][18].

La figure I.1 illustre un DODAG dans une instance RPL et la figure I.2 représente la construction d'un DODAG.

7.2. Types de nœuds

RPL définit trois types de nœud : LBR, Router et Host.

- **LBR (Low Power and Lossy Border Router) :** C'est la racine du DODAG qui représente un point de collecte dans le réseau. Il peut également jouer le rôle d'une passerelle (ou un routeur de périphérie) entre Internet et le LLN[3].
- **Router :** Il représente l'équipement chargé de transférer et de générer du trafic[3].
- **Host :** Il fait référence à un équipement final capable de générer du trafic de données,

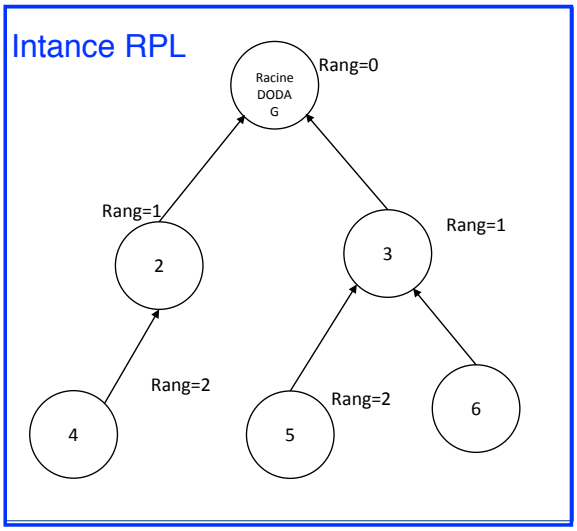

FIGURE I.1 – DODAG dans une instance RPL

mais qui n'est pas capable de le relayer[3].

7.3. Messages de contrôle du protocole

Pour construire et maintenir à jour un DODAG, RPL utilise des messages de contrôle ICMPv6 (Internet Control Message Protocol Version 6). Les messages de contrôle les plus importants sont :

- **DIO (DODAG Information Object) :** il contient les informations requises par un nœud pour découvrir une instance RPL, obtenir les paramètres de configuration, et sélectionner un ensemble parent dans le DODAG[17]. Il est envoyé en multicast à toute la structure du DODAG.
- **DIS (DODAG Information Solicitation) :** ce message de contrôle est envoyé pour solliciter un DIO à partir d'un nœud. Lorsqu'un nouveau nœud désire se rattacher à un DODAG déjà formé, il diffuse un message DIS pour obtenir des informations de configuration de son voisinage.
- **DAO (DODAG Advertisement Object) :** ces messages servent à construire les routes du trafic descendant contrairement aux messages DIO qui sont utilisés pour construire les routes pour le trafic ascendant.
- **DAO-ACK :** ce message est envoyé en unicast pour acquitter un message DAO (parent DAO ou racine DODAG).

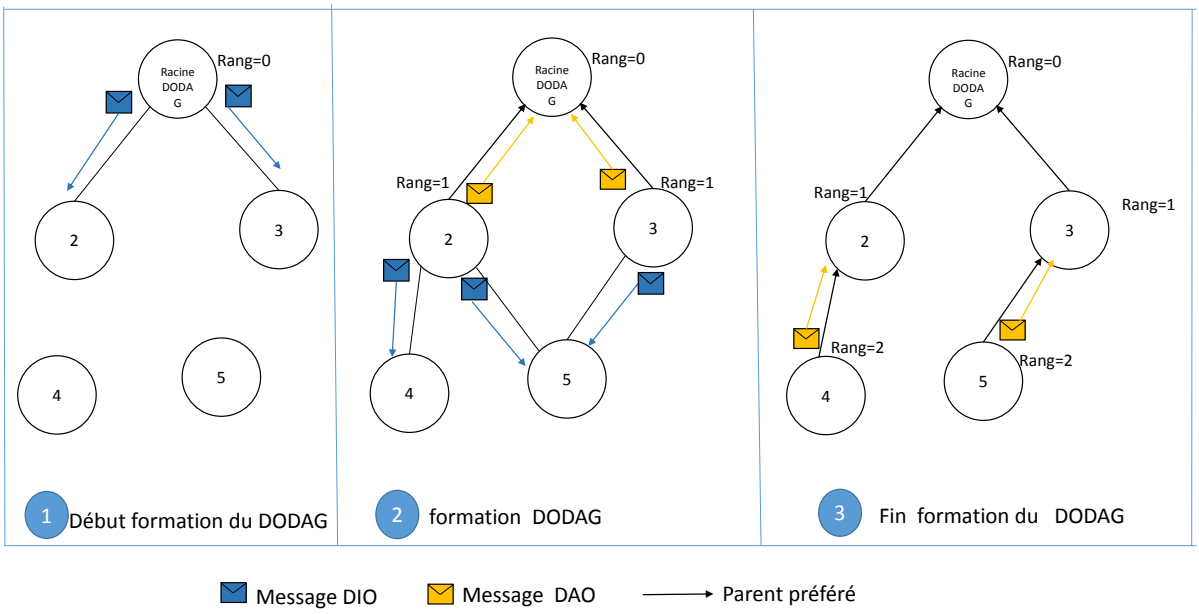

FIGURE I.2 – Construction du DODAG

7.4. Timer Trickle

Le protocole RPL étant proactif, les messages DIO sont envoyés périodiquement afin de configurer et de mettre à jour les routes. Le timer Trickle est utilisé par RPL pour minimiser l'envoie de ces derniers. L'algorithme Trickle permet donc aux nœuds du réseau LLN d'échanger des DIO de manière très robuste, économe en énergie, et simple[18]. La fréquence d'envoie de DIO augmente avec la détection d'incohérence dans le réseaux (afin de réagir rapidement aux éventuelles pannes de nœud ou de la liaison) et diminue au cas où le réseau est stable.

L'algorithme Trickle se déroule en six (6) étapes[18] :

1. Initialiser un intervalle de temps I dans la plage **[Imin, Imax]**
2. Initialiser un compteur **c=0** et et choisir aléatoirement un instant **t** entre **[I/2, I]**
3. A chaque réception d'un message de configuration cohérent, incrémenter **c**.
4. A l'instant **t**, transmettre les messages de contrôles si et seulement si **c < k** (k est une constante de redondance > 0)
5. Lorsque l'intervalle *I* expire, doubler la taille de l'intervalle et revenir à l'étape 2.
6. Si un message incohérent détecté, réinitialiser l'intervalle **I** à **Imin** et recommencer à l'étape 2

7.5. Modèles de communication

Le protocole de routage RPL définit trois modèles de communication : P2MP (Point-to-multipoint), MP2P (Multipoint-to-point) et P2P (Point-to-point). La figure I.3 illustre les modèles de communication.

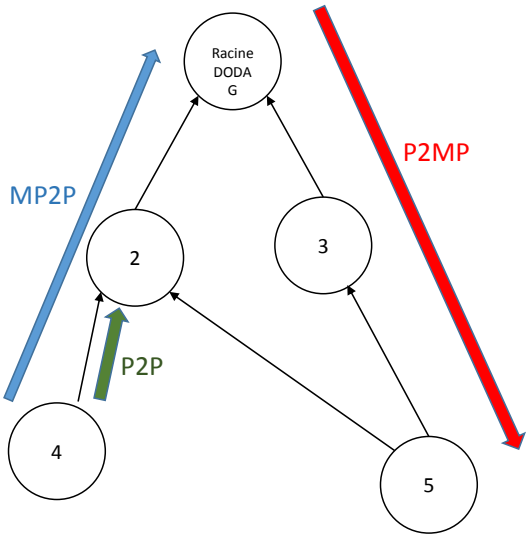

FIGURE I.3 – Modèles de communication

Dans les lignes qui suivent, nous expliquerons chacun des modèles.

7.5.1 Point-to-multipoint (P2MP)

Dans ce mode, la communication est descendante. Le modèle Point-to-multipoint est utilisé lorsque la racine du DODAG transmet des messages aux autres nœuds du réseaux.

7.5.2 Multipoint-to-point (MP2P)

Dans ce modèle, la communication est ascendante. Autrement dit, les informations partent des autres noeuds du DODAG vers le nœud racine.

7.5.3 Point-to-point (P2P)

Ce modèle est utilisé lorsque les nœuds communiquent entre eux en échangeant des informations sans que le root ne soit ni émetteur ni destinataire du message.

7.6. Modes d'opération

Le protocole RPL définit en fonction de la capacité des nœuds en terme de mémoire, et puissance de calcul, deux modes d'opération : le Storing Mode et le Non-Storing mode.

7.6.1 Storing Mode

Dans ce mode, chaque nœud a sa propre table de routage contenant les chemins vers ses nœuds fils et ses nœuds parents. Lorsqu'un nœud émetteur transmet un message à un nœud récepteur, le message monte jusqu'au premier ancêtre (premier parent) commun aux deux nœuds, puis redescend vers le nœud destinataire.

7.6.2 Non-Storing Mode

Dans ce mode, les nœuds intermédiaires ne stockent pas les informations de routage car ces nœuds ne possèdent pas suffisamment de ressources pour le faire. Elles sont stockées par le nœud racine seulement. Les autres nœuds du réseau conservent uniquement les adresses de leur parent direct. Lorsqu'un nœud désire transmettre un message à un autre, il l'envoie à la racine du DODAG qui, par la suite, achemine le message vers le destinataire.

7.7. Métrique

Deux types de métriques de routage ont été définis par le protocole RPL pour le calcule des chemins : métrique de nœud qui considère les attributs relatifs à un nœud et la métrique de liaison qui prend en compte les attributs de liens[19].

- **Métrique de nœud :**
  - **Les états ou attributs** qui fournissent des informations sur les caractéristiques du nœud telle que l'usage du CPU, la mémoire consommée ;

— **Le nombre de sauts** qui équivaut au nombre de nœud traversé pour atteindre la racine. Plus le nombre de sauts est faible, plus le nœud est proche de la racine ;

— **L'énergie** qui représente la source d'énergie du nœud (batterie, secteur, etc.).

- **Métrique de liaison :**

— **Le débit** qui représente la quantité de données passant par un lien en une unité de temps ;

— **La latence** qui exprime la durée de la transmission d'un paquet de l'émetteur au récepteur ;

— **La fiabilité** qui est une représentation abstraite pour exprimer la qualité du lien, soit l'ETX (nombre de retransmission attendue). L'ETX est le nombre de transmissions qu'un nœud s'attend à effectuer vers une destination afin de livrer avec succès un paquet. Plus il y'a perte de paquet, plus la valeur de l'ETX augmente et inversement plus le taux de délivrance avec succès augmente et plus la valeur d'EXT diminue.

— **La couleur de lien** qui peut être une représentation de différents types de liens par des valeurs abstraites. Elle est utilisée pour éviter ou attirer des liens spécifiques pour des types de trafic spécifiques.

7.8. Fonction objective

Une fonction objective définit les métriques et les contraintes permettant aux nœuds de calculer leur rang, d'établir une liste de parent et de sélectionner un parent préféré dans cette liste. Le groupe de travail ROLL a défini deux fonctions objectives :

7.8.1 OF0 (Objective Function Zero)

Elle est définie dans la RFC 6552 et est conçue pour trouver le plus court chemin vers la racine . Elle implémente le nombre de sauts comme métrique, tous les liens participent avec le même poids pour la sélection du chemin[21]. OF0 sélectionne le voisin avec le meilleur rang comme parent préféré et un autre nœud comme secours du parent préféré (pour une tolérance aux pannes). Tout le trafic ascendant reçu par un nœud est acheminé vers la racine en utilisant le parent préféré ou le nœud de secours, selon les conditions de liaison. OF0 ne tente d'effectuer

aucun équilibrage de charges. Le nœud secours est utilisé pour transmettre un paquet au nœud racine lorsque le parent préféré ne peut pas être utilisé.

7.8.2 MRHOF (Minimum Rank with Hysteresis Objective Function)

MRHOF a été développé pour sélectionner le plus court chemin tout en réduisant les changements excessifs de parent préféré. Pour cela, il utilise deux mécanismes : le premier cherche à trouver le plus court chemin tandis que le second effectue l'hystérésis. L'hystérésis consiste à élire un nœud comme parent si et seulement s'il est le plus court chemin d'au moins un seuil. Par défaut, MRHOF utilise la métrique ETX pour calculer le coût de chaque chemin. Cependant, il peut utiliser n'importe quelle métrique de routage décrit dans [20]. Il a été spécifié dans la *RFC 6719* .

7.9. Évitement et détection de boucle

Les boucles sont un problème rencontré fréquemment dans les protocoles de routage à vecteur de distance. Si une boucle existe dans le DODAG, cela consommera davantage les ressources (mémoire, traitement et énergie) des nœuds concernés. Le réseau risque une indisponibilité si les nœuds formant un circuit sont les seuls chemins vers la racine du DODAG. RPL intègre des mécanismes pour détecter et éviter les boucles. Pour cela, il s'appuie sur le rang et sur un autre mécanisme qui est celui de la validation des routes.

Une boucle peut subvenir pour plusieurs raisons. Par exemple, lorsqu'un nœud A perd tous ses parents, il augmente son rang puis sélectionne son enfant B comme nouveau parent. L'enfant B sélectionne A comme parent car il ne peut pas se rattacher à un autre et ainsi de suite.

Pour éviter les boucles, le protocole RPL limite la valeur de rang maximum autorisée dans un graphe. Cette règle évite le comptage à l'infini lorsqu'une boucle est formée.

Le nœud sortant peut utiliser un mécanisme de détachement, qui consiste à former un DODAG intermédiaire et à rejoindre le DODAG principal ultérieurement[3][22].

RPL peut également détecter les boucles en utilisant le mécanisme de validation de chemin de données comme évoquer précédemment. Cela consiste à inclure les informations de routage (données de contrôle) dans les paquets de données. Les informations de contrôle sont

transportées dans les paquets de données via des flags placés dans l'en-tête IPv6 :

- **Le flag 'O ' :** indique la direction attendue du paquet, c'est à dire, vers le haut ou le bas. Si un nœud place ce flag à 1, le paquet est destiné à un descendant, sinon le paquet est supposé être envoyé à un parent avec un rang inférieur, vers la racine du DODAG[17].
- **Le flag 'R' :** il indique qu'une erreur de rang a été détectée par un nœud transférant le paquet. Ce flag est mis à 1 lorsqu'un nœud observe une incohérence entre la direction supposée du paquet indiquée par le flag 'O' et le rang du nœud qui vient de le transférer[17].

7.10. Maintenance de la topologie

Dans un protocole de routage dynamique, il est important d'utiliser des mécanismes de réparation afin de maintenir la topologie en cas de défaillance des nœuds et/ou des liaisons. RPL prend en charge deux mécanismes de réparation :

7.10.1 Réparation locale

Elle est déclenchée par un nœud lorsqu'il détecte une incohérence dans le réseau (une boucle locale par exemple). Le nœud affecté trouve une route de secours sans essayer de réparer la DODAG complètement. Ce mécanisme de réparation locale permettra au réseau de converger à nouveau dans un délai raisonnable.

7.10.2 Réparation globale

La réparation globale est initiée par le nœud racine du DODAG. Pour cela, il incrémente le numéro de la version du DODAG ce qui entraîne la construction d'une nouvelle topologie en autorisant tous les nœuds du réseau à pouvoir choisir un nouveau rang dans le DODAG sans être contraint par leur rang dans l'ancienne version du DODAG. Cette réparation optimise le graphe mais elle est coûteuse et moins rapide par rapport à la réparation locale.

7.11. Sécurité

Le protocole RPL fournit des mécanismes de cryptographie facultatifs pour sécuriser ses messages de contrôle et pour assurer la confidentialité, l'intégrité et l'authenticité du réseau.

Il propose trois modes de sécurité de base[17] :

- **Non sécurisé :** dans ce mode, les messages de contrôle RPL sont envoyés sans aucun mécanisme de sécurité. D'autres mécanismes de sécurité peuvent être utilisés (mécanismes de sécurité de la couche liaison ou de la couche transport par exemple).
- **Pré-installé :** il consiste à chiffrer les messages à l'aide de clés pré-installées sur les nœuds. Ainsi, les nœuds rejoignant une instance disposent de clés pré-installées qui leur permettront d'envoyer et de recevoir des messages sécurisés.
- **Authentifié :** dans ce mode, les nœuds ont des clés pré-installées comme dans le mode pré-installé. Cependant, si un nœud veut participer en tant que routeur il doit obtenir une autre clé d'une autorité d'authentification. La clé pré-installée lui permet uniquement de rejoindre une instance en tant que feuille.

8. Conclusion

RPL est un protocole de routage à vecteur de distance spécialement conçu par l'IETF pour répondre aux besoins des réseaux LLN. Dans cette partie, nous avons présenté brièvement les réseaux LLN en abordant leurs multiples domaines d'utilisation et leurs contraintes de routage. Ensuite nous avons détaillé le fonctionnement de son protocole de routage en décrivant les messages de contrôle, les mécanismes de construction et maintenance de la topologie, et les modèles de communication définis par le protocole en question.

Dans le chapitre suivant, nous aborderons la sécurité dans les réseaux LLN.

Chapitre II

Securité des réseaux LLN

1. Introduction

Les LLN sont des réseaux fortement contraints en terme de puissance de traitement, d'énergie, et de mémoire. Ils sont caractérisés par des liaisons instables avec des taux de perte élevés. Ces contraintes rendent les LLN vulnérables à plusieurs types d'attaques. Les mécanismes de sécurité définis par RPL tels que la réparation de la topologie, l'évitement et la détection de boucle, et les mécanismes de sécurité de base garantissent le fonctionnement du protocole et aident à contrer certaines des attaques. Mais, la topologie RPL peut subir des attaques provenant des nœuds internes. De plus, un intrus peut écouter le réseau ou exploiter des vulnérabilités connues, puis accéder à une clé partagée ou contourner les protections traditionnelles basées sur le chiffrement.

Dans ce chapitre, nous abordons les différentes classifications des attaques que subit RPL. Puis nous étudions les attaques de topologie (principalement les attaques de rang). Nous nous intéresserons ensuite aux mécanismes utilisés dans la littérature pour atténuer les impacts des attaques.

2. Classification des attaques

Dans la littérature, il existe plusieurs types de classifications d'attaques. Nous décrivons quelques unes ci-après.

2.1. Classification des attaques basée sur les comportements

Dans cette classification, les attaques sont groupées selon leur comportement. A cet effet, les attaques peuvent être réalisées afin d'effectuer une modification (attaque active) ou non (attaque passive) [11].

Dans le cas d'une attaque passive, l'intrus va tenter de dérober les informations par audit, ce qui rend sa détection relativement difficile. En effet, cet audit ne modifie pas les données, ni n'altère le comportement du réseau. Généralement, les réseaux sans fil sont les plus vulnérables aux attaques passives. Les attaquants (généralement cachés) exploitent les lignes de communication pour collecter des données.

Contrairement aux attaques passives, les attaques actives impliquent la modification ou la création des messages (non légitimes) et la perturbation du bon fonctionnement du réseau. Les dysfonctionnements des nœuds et les altérations des comportements de ces derniers sont considérés comme des attaques actives. Dans cette catégorie d'attaque, les actes malveillants sont menés non seulement contre la confidentialité des données, mais aussi contre leur intégrité. Les attaques actives peuvent être détectées (car les performances du réseau peuvent être dégradées à la suite de ces attaques).

2.2. Classification des attaques basée sur les critères de sécurité

Cette classification regroupe les attaques en se basant sur le modèle de référence de sécurité CIA (Confidentialité, Intégrité, Disponibilité).

- La confidentialité signifie la protection des informations de routage et des échanges contre la lecture non autorisée. Les attaques contre la confidentialité sont généralement des écoutes clandestines.
- L'intégrité implique la protection des informations de routage contre toute modification non autorisée.
- La disponibilité consiste à garantir que les échanges d'informations de routage sont accessibles par les nœuds. Les attaques de cette catégorie tentent d'impacter systématiquement la disponibilité du réseau. Autrement dit, les attaques de cette catégorie visent à mettre en péril les ressources du réseau (batterie, mémoire, traitement, disponibilité des liens).

2.3. Classification des attaques basée sur les éléments du réseaux RPL impactés

Cette classification couvre les attaques de routage contre le protocole RPL (en tenant compte des objectifs de l'attaque et des éléments du réseau RPL qui sont impactés). D'après elle, RPL peut subir trois grandes catégories d'attaque [6].

- La première catégorie regroupe les attaques qui visent à épuiser les ressources du réseau (énergie, mémoire et puissance de traitement). Ces attaques peuvent être dangereuses pour les réseaux contraints car elles tentent de réduire la durée de vie des équipements et de dégrader les performances du réseau en surchargeant les nœuds ou en générant des trafics très importants. Ces attaques conduisent souvent à une congestion du réseau et également à la saturation des nœuds RPL.
- La deuxième catégorie correspond aux attaques contre le trafic réseau, telles que les attaques par écoute clandestine ou les attaques de détournement qui usurpent l'identité des nœuds légitimes.
- La troisième catégorie est composée des attaques qui ciblent la topologie du réseau RPL. Elles perturbent le fonctionnement du réseau et ouvrent l'accès à plusieurs autres attaques. Ces attaques sont considérées comme étant extrêmement dangereuses. Cette catégorie d'attaque fait l'objet de notre étude. Nous nous intéressons principalement aux attaques contre le rang.

3. Impact de l'attaque de rang sur la topologie du protocole RPL

Le rang est indispensable dans la construction et le maintien de la topologie du protocole RPL. Concrètement, dans un réseau RPL, le rang est associé à chaque nœud et correspond à la position du nœud dans le graphe par rapport au nœud racine. Il est utilisé pour l'optimisation des itinéraires, la prévention des boucles etc. Comme mentionné précédemment, le rang des nœuds augmente toujours dans le sens descendant afin de préserver la structure acyclique du DODAG en d'autres termes un nœud doit choisir un rang supérieur à celui de son parent.

Étant donné que le rang joue un rôle crucial dans le fonctionnement de RPL et qu'il est

quasiment lié à toutes les opérations du protocole en question, toute attaque visant le rang peut également avoir des impacts multiples sur les performances de RPL.

Une étude sur l'impact de l'attaque de rang a été effectuée dans [9]. Les résultats ont révélé que cette attaque peut avoir un impact grave sur les performances du réseau, en particulier lorsqu'elle est mise en œuvre dans une zone où plusieurs nœuds légitimes sont localisés.

4. Attaque du pire parent

Cette attaque consiste à choisir systématiquement le plus mauvais parent comme parent préféré en tenant compte de la fonction objective. En effet, le nœud malveillant envoie son rang à ses voisins. Ensuite, il sélectionne son pire parent (celui avec le rang le plus élevé) pour transmettre les paquets des nœuds enfants. Il en résulte que le chemin obtenu n'est pas optimisé (retard lors de la transmission des paquets), ce qui entraîne des dégradations de performance. Cette attaque ne peut pas être facilement détectée car le nœud enfant compte sur son parent pour acheminer les paquets. Elle ne peut être surveillée par les voisins. La figure II.1 illustre une attaque du pire parent dans un DODAG.

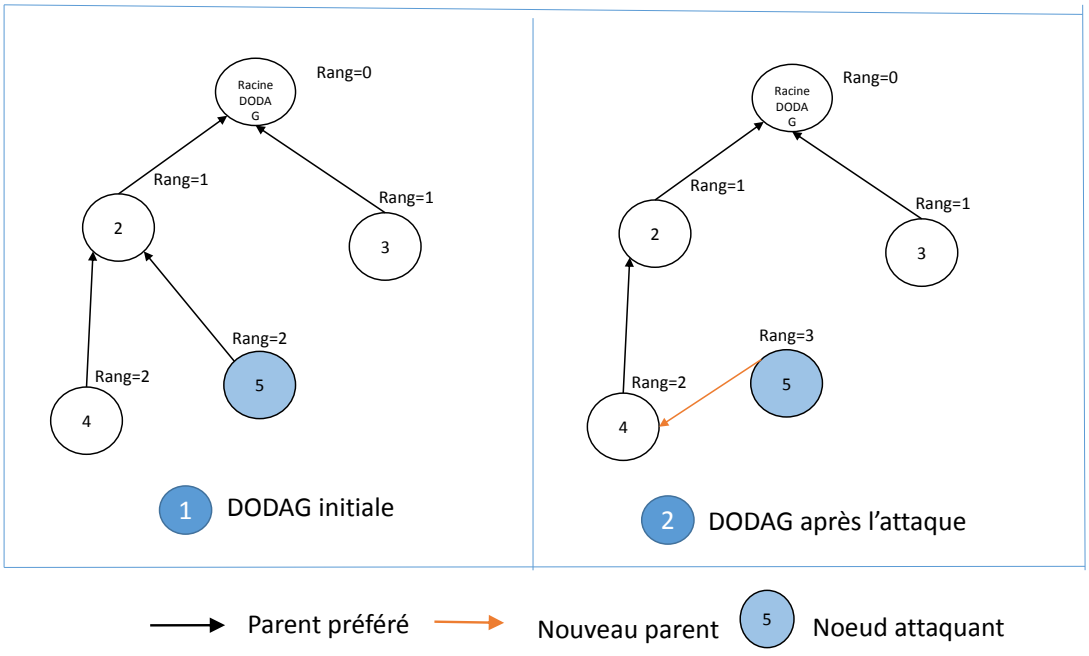

FIGURE II.1 – Attaque du pire parent

5. Attaque par augmentation de rang

L'attaque par augmentation de rang a pour objectif de générer des boucles dans le réseau, de rendre le graphe (DODAG) non optimal ou d'isoler complètement un ensemble de nœuds du réseau.

Un nœud malveillant, appelé A par exemple, peut annoncer une valeur de rang supérieure à celle qu'il est supposé avoir. Des boucles sont formées lorsque son nouveau parent préféré, appelé B par exemple, était dans son sous-DODAG antérieur et uniquement si l'attaquant n'utilise pas de mécanismes d'évitement de boucle.

Le nœud B tentera alors de trouver un autre parent préféré qui lui permettra d'avoir une meilleure position dans le DODAG (rang optimal). S'il n'a que A comme parent, alors le sous-DODAG composé de A et B contient une boucle. Ce qui perturbera la topologie.

La figure II.2 illustre une attaque par augmentation de rang dans un DODAG.

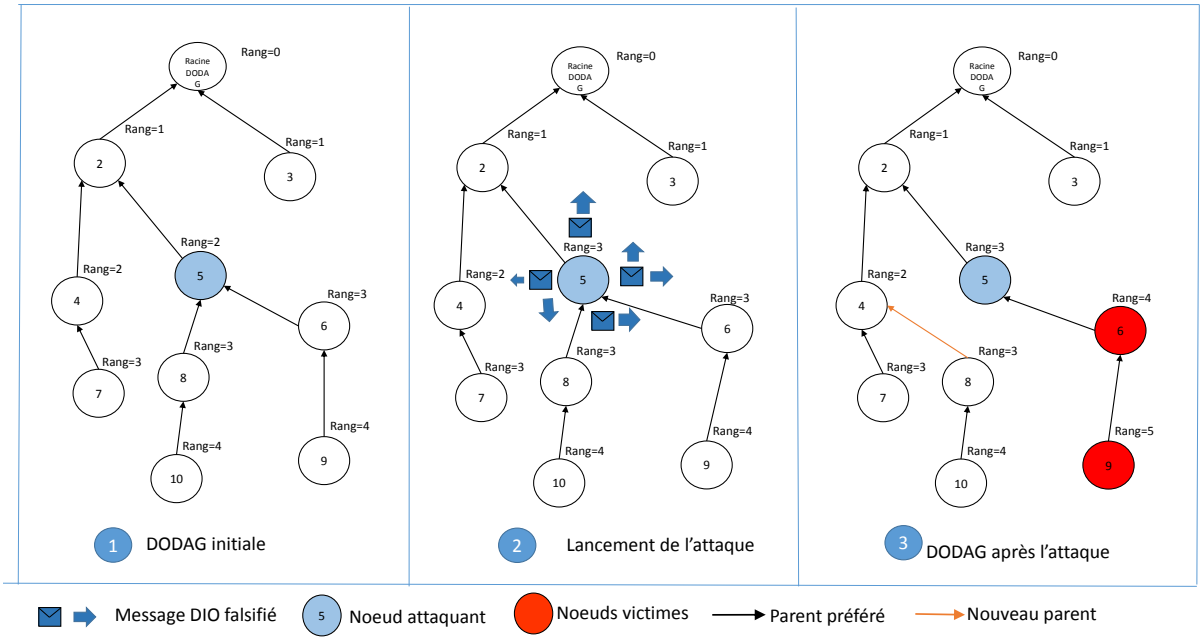

FIGURE II.2 – Attaque par augmentation de rang

6. Attaque par diminution de rang

Dans une attaque par diminution de rang, un nœud malicieux annonce illégalement un rang inférieur de sorte qu'il se localise plus près de la racine. Puisque dans un graphe DO-DAG, plus le rang est bas, plus le nœud est proche de la racine alors plusieurs nœuds voisins sélectionnent le nœud malicieux comme parent préféré pour se connecter à la racine. Donc les communications de plusieurs nœuds seront attirées vers le nœud en question.

Cette attaque est particulièrement dangereuse et probablement la plus critique contre la topologie car plusieurs autres attaques peuvent être réalisées à partir d'elle. Le nœud qui est à l'origine de l'attaque peut alors décider de filtrer le trafic passant par lui en routant ou en bloquant certains paquets de données. Il peut manipuler le trafic et isoler complètement un ensemble de nœuds du réseau en d'autres termes les nœuds victimes ne reçoivent aucune donnée provenant des autres nœuds y compris le nœud racine.

La figure II.3 illustre une attaque par diminution de rang dans un DODAG.

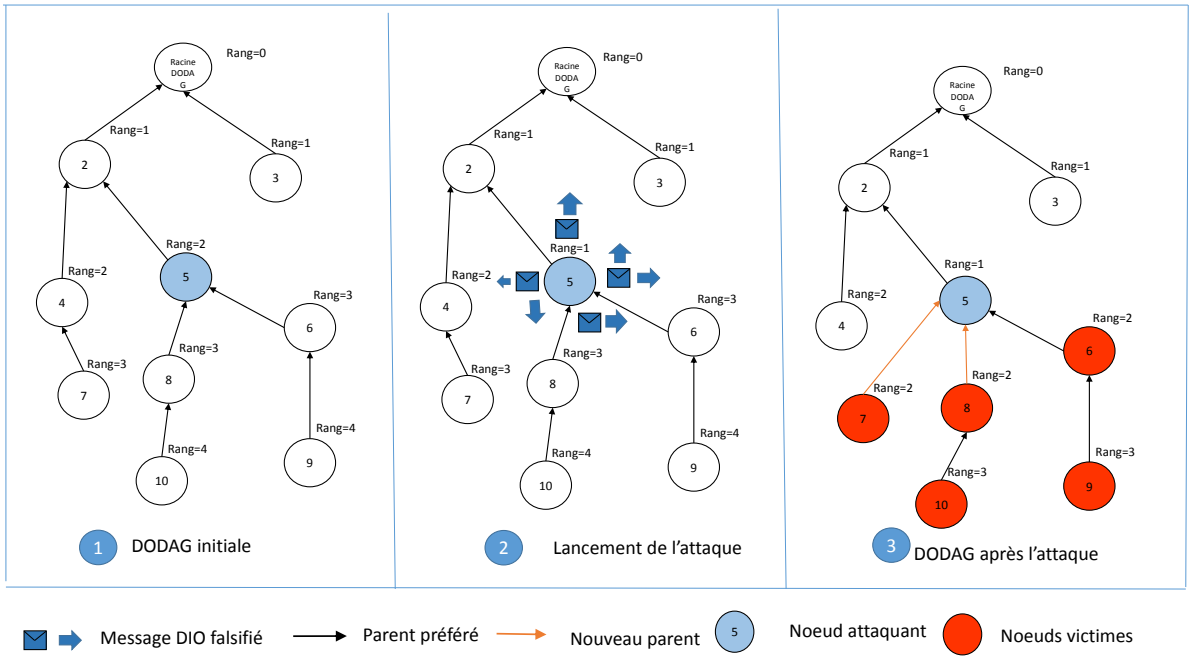

FIGURE II.3 – Attaque par diminution de rang

7. Contre-mesures

Plusieurs mesures peuvent être envisagées pour protéger les réseaux RPL contre des attaques. Des mécanismes peuvent être utilisés pour assurer :

- La fiabilité des données échangées entre les nœuds lors de la synchronisation de leurs paramètres de routage.
- L'intégrité des données
- L'authentification des nœuds (identité des nœuds) etc.

Cependant, parfois même avec des données fiables et des nœuds authentifiés, il existe des attaques qui ciblent le manque de ressources et d'énergie pour le traitement des données dans les réseaux LLN, elles visent donc à inonder le réseau de données inutiles et à le mettre hors service.

7.1. Protection d'un réseau

En informatique, la protection des réseaux contre les attaques passe par trois éléments essentiels : la prévention, la détection, l'atténuation [11].

- La prévention vise à prévenir les attaques avant qu'elles ne surviennent. Les mécanismes de prévention des intrusions peuvent résister aux attaquants externes envers l'IoT, mais ils ne sont pas spécifiquement conçus pour résister aux attaquants internes.
- La détection consiste à repérer les intrusions qui n'ont pas pu être prévenues. Dans le contexte des LLN, cette action consiste à identifier les nœuds qui sont compromis. La seule façon de réagir contre les attaques en cours, en particulier les attaques internes, est d'utiliser les systèmes de détection d'intrusion (IDS). Une fois qu'une intrusion est détectée, un mécanisme d'atténuation est employé pour minimiser les effets néfastes de l'attaque en cours.
- L'atténuation a pour objectif de réduire les impacts des attaques après une intrusion. Dans les LLN, par exemple, les nœuds compromis peuvent être exclus d'un réseau (aucun trafic ne passe par ce dernier).

7.2. Système de détection d'intrusion

Un système de détection d'intrusion (IDS) est un outil ou un mécanisme utilisé dans le but de détecter des attaques contre un système ou un réseau. Une fois qu'une attaque est détectée par un IDS, il peut déclencher une alerte. Les IDS s'appuient généralement sur les signatures numériques (emprunte d'une entité malveillante) et sur les anomalies pour reconnaître une tentative d'intrusion [5].

La reconnaissance basée sur les signatures repose sur une base de connaissance contenant les signatures des attaques connues. Cette technique est statique et ne peut détecter que les attaques dont les signatures ont été ajoutée manuellement dans l'IDS.

La reconnaissance basée sur les anomalies repose sur l'analyse du système. L'IDS compare le comportement du système (à un instant) à un modèle de référence (le comportement ordinaire) pour repérer une tentative d'intrusion. Cette technique a la capacité de détecter presque toutes les attaques et de s'adapter à de nouveaux environnements. Mais, avec cette approche, l'IDS peut déclencher une alarme lorsqu'il n'y a pas d'attaque et rester silencieux en cas d'attaque.

Une autre reconnaissance, consiste à combiner les techniques basées sur les signatures et les techniques basées sur les anomalies. Cette approche essaye d'équilibrer le coût de stockage de la détection basée sur la signature et le coût de calcul des techniques basées sur les anomalies.

7.3. VeRA (Version Number and Rank Authentication)

Les mécanismes de sécurité basés sur la cryptographie sont utilisés en première défense, ils n'empêchent que les attaques externes. Lorsqu'un nœud est compromis, il devient par conséquent un attaquant interne, les techniques cryptographiques deviennent inutiles et ne peuvent plus protéger le réseau [14].

VeRa est un protocole qui détecte les attaques contre le numéro de version et contre le rang en utilisant des chaînes de hachage unidirectionnelles. Il empêche les nœuds compromis d'envoyer un message DIO avec un rang falsifié ou de se faire passer pour une racine DODAG pour changer le numéro de version [15][14] [10].

Le principe de VeRA repose sur deux propriétés clés des chaînes de hachage. Première-

ment, un nœud qui contient un élément de hachage ne peut pas calculer l'inverse. Deuxièmement, il est facile de calculer la fin de la chaîne si la longueur est connue.

Il utilise un algorithme de cryptage asymétrique (RSA[2], par exemple) pour garantir l'authenticité des messages. La clé secrète est connue par la racine du DODAG seulement et la clé publique est connue par tous les autres nœuds [15].

7.3.1 Problèmes de VeRa

En plus d'avoir une complexité de calcul élevée (dûe aux opérations de hachage), VeRA reste vulnérable aux attaques de topologie par usurpation de rang. La première vulnérabilité permet à un attaquant de créer une chaîne de hachage de rang falsifiée et de revendiquer n'importe quel rang dans la topologie. Et la seconde permet à un attaquant de rejouer le rang de son parent et monter ainsi d'un niveau dans la hiérarchie [10][12].

7.4. SVELTE

SVELTE est un IDS, proposé dans [5], qui surveille le réseau et détecte les incohérences qui y sont liées. Il peut identifier les nœuds malicieux et demander à leurs voisins de les mettre sur liste noire, ce qui empêche les intrus de participer à la topologie RPL active. Dans SVELTE, les techniques de détection ciblent principalement les attaques de topologie. Cependant, SVELTE est extensible et peut être utilisé pour détecter d'autres attaques[5]. Il est adapté aux fortes contraintes des LLN.

7.4.1 Description

SVELTE dispose de trois modules principaux [5] :

- Le premier module, appelé 6LoWPAN Mapper (6Mapper), rassemble des informations sur le réseau RPL et reconstruit le réseau. Pour cela, tous les noeuds du réseau envoient au sink des paquets contenant leurs informations (identifiant, rang, parent préféré etc.) et celles de leur voisinage (identifiant et rang de tous les nœuds voisins) ;
- Le deuxième module est le composant de détection d'intrusion qui analyse les données cartographiées et détecte l'intrusion. Plusieurs algorithmes ont été implémentés dans ce

module afin de détecter une attaque contre la topologie ;

- Le troisième module, un mini-pare-feu distribué, est conçu pour décharger les nœuds en filtrant le trafic indésirable avant qu'il n'entre dans le réseau à ressources limitées.

7.4.2 Problèmes de SVELTE

SVELTE a plusieurs problèmes notamment :

- Il a un taux élevé de fausses alertes ;
- La racine du DODAG doit donner des informations concernant les intrus à tous les autres nœuds ;
- Il surcharge le trafic à cause des paquets d'informations qu'envoient les nœuds au 6Mapper ;
- Il nécessite des ressources supplémentaires dans les noeuds (0.365k de RAM et 1.76k de ROM)[5].

7.5. LEADER (Low Overhead Rank Attack Detection for Securing RPL based IoT)

LEADER est une solution qui détecte les attaques par diminution et par augmentation de rang dans les réseaux LLN. Il est composé de deux modules :

- Un premier module implémenté dans tous les nœuds mis à part le sink ;
- Un second module centralisé implémenté uniquement dans le sink.

Contrairement à svelte, LEADER n'utilise pas de paquet de contrôle supplémentaire. Il modifie la structure du message de contrôle DAO en ajoutant un champ de 16 bits contenant le rang d'un nœud et un deuxième champ pour celui de son parent préféré. Ainsi le sink collecte les informations du réseau, les analyse puis vérifie qu'aucun des nœuds ne modifie illégalement son rang.

Et pour assurer l'authenticité et l'intégrité des DAO transmis, il utilise une fonction cryptographique de hachage appelé LOCHA.

7.5.1 Problèmes de LEADER

Tout comme SVELTE, LEADER est une solution qui dépend du nœud central (sink). Si le sink tombe en panne, la solution risque de dysfonctionner. En outre, il utilise une fonction cryptographique de hachage qui consomme les ressources des noeuds.

Le tableau II.1 résume les attaques de rang et leurs contre-mesures.

TABLE II.1 – Table résumant les attaques de rang et leurs contre-mesures [6]

Attaque	Contre-mesures
Attaque par diminution de rang	VeRA, SVELTE, LEADER
Attaque par augmentation de rang	LEADER, Mécanismes de détection et d'évitement de boucle RPL
Attaque du pire parent	SVELTE

8. Conclusion

Les attaques de rang peuvent créer des chemins non optimisés, plus de surcharge et plus de collisions de paquets, dégradant ainsi les performances du réseau. Dans ce chapitre, nous avons étudié différents types d'attaques de rang qui peuvent dégrader les performances de RPL. Puis nous avons abordé leur impact et les mécanismes utilisés pour atténuer leur effet. Dans le chapitre suivant, nous proposerons notre solution pour détecter des attaques par diminution et par augmentation de rang et mitiger de leurs impacts.

Chapitre III

Solution proposée pour les attaques de rang

1. Introduction

Dans la littérature, plusieurs solutions centralisées existent pour détecter les attaques de rang. Avec ces solutions, les sink ont une cartographie complète (vue globale) du réseau qu'ils établissent à partir des paquets de contrôle reçues des autres nœuds du réseau. Ces Paquets génèrent un trafic important, alors que les réseaux LLN sont caractérisés par des liens instables à débit limité. En outre, il suffit juste que le nœud central soit corrompu ou tombe en panne pour que les contres-mesures cessent de fonctionner.

Nous proposons dans ce chapitre une solution distribuée pour les réseaux LLN ayant pour objectif de détecter les attaques par augmentation ou par diminution de rang et de réagir afin de mitiger leurs impacts. Nous avons opté pour une solution distribuée car elle permet un équilibrage de charges et ne dépend pas d'un seul nœud central. Tous les nœuds collaborent pour détecter les attaques.

2. Description de la solution proposée

Notre solution est basée sur la surveillance des nœuds du réseau. Chaque nœud participe à la détection d'attaques en surveillant ses enfants. Lorsqu'un nœud N calcule son rang, il applique la formule suivante :

$$\mathscr{R}_N = \mathscr{R}_P + cout\_de\_chemin \tag{III.1}$$

avec :

$\mathscr{R}_N$ est le rang du nœud N

$\mathscr{R}_P$ est le rang de son parent préféré P

Le cout_de_chemin est calculé en utilisant L'ETX.

Nous nous appuyons sur cette propriété pour détecter les attaques. Les parents préférés contrôlent périodiquement le rang de leurs fils et vérifient que ces derniers ont annoncé un rang cohérent avec la formule. La métrique que nous utilisons est l'ETX et chaque nœud doit l'utiliser pour calculer son rang.

2.1. Détection d'attaque par augmentation de rang

Après la formation du DODAG, le nœud malveillant lance une attaque en incrémentant illégalement et périodiquement son rang. A chaque augmentation de rang, les nœuds de sa descendance recalculent leur rang.

Chaque parent préféré estime le rang de ses fils et déclare qu'un d'eux est attaquant qui augmente son rang si :

$$\mathscr{R}_N > \mathscr{R}_P + Z \tag{III.2}$$

avec

$\mathscr{R}_N$ est le rang annoncé par un nœud N

$\mathscr{R}_P$ est le rang de N estimé par son parent préféré P

Z est une constante représentant la valeur maximale entre $\mathscr{R}_N$ et $\mathscr{R}_P$

2.2. Détection d'attaque par diminution de rang

Pour détecter cette attaque, chaque parent préféré estime le coût de chemin qui le sépare de chacun de ses fils. Si un des fils diffuse un rang inférieur d'un seuil au rang estimé par son parent préféré, il sera déclaré comme attaquant dans le réseau.

Un nœud N est déclaré attaquant qui diminue son rang si :

$$\mathscr{R}_N < \mathscr{R}_P - K \tag{III.3}$$

avec

$\mathscr{R}_N$ est le rang annoncé par un nœud N

$\mathscr{R}_P$ est le rang de N estimé par son parent préféré P

et K est une constante représentant la valeur maximale entre $\mathscr{R}_N$ et $\mathscr{R}_P$

2.3. Mitigation de l'impact des attaques

Après la détection d'une attaque, l'étape qui suit consiste à prendre des mesures afin de contrer les attaques ou du moins à réduire leurs impacts. Pour cela, lorsqu'une attaque est détectée, tous les nœuds du réseau sont alertés. Autrement dit, le nœud qui détecte une attaque diffuse dans le réseau les informations de l'attaquant. Ainsi les nœuds du réseau filtreront l'attaquant et essayeront de choisir un autre parent.

3. Algorithmes

Nous avons utilisé les algorithmes 1, 2 et 3 dans notre solution.

Algorithme 1 : Fonction Detecter attaque()

```
Debut
    Pour chaque N dans Table_de_routage Faire
        Si N est mon fils direct Alors
            // Rpp est le rang du parent préféré de N
            // etx est l'etx estimé de N
            // Rn est le rang de N
            Si est_attaquant_increased (Rn,Rpp,ext) ou est_attaquant_decreased (Rn,Rpp,ext) Alors
                Si N existe dans la liste de suspect Alors
                    ajouter N à la liste noire
                    transmettre les informations de N
                Sinon
                    ajouter N à la liste de suspect fin
                Fin si
            Fin si
        Fin si
    Fin
Fin
```

Algorithme 2 : Fonction est_attaquant_increased

Entrées : RangParentPréféré, RangN, ETXestimé
Sorties : reponse
Debut

```
Si     RangN > RangParentPrefere + ETXestime + Z Alors
    reponse←Vrai
    retourner reponse
Sinon
    reponse ← Faux
    retourner reponse
Finsi
Fin
```

Algorithme 3 : Fonction est_attaquant _decreased

Entrées : RangParentPréféré, RangN, ETXestimé
Sorties : reponse
Debut

```
Si  RangN < RangParentPrefere + ETXestime - K Alors
    reponse ← Vrai
    retourner reponse
Sinon
    reponse ← Faux
    retourner reponse
Finsi
Fin
```

4. Critères d'évaluation de la solution

Nous utilisons plusieurs critères pour évaluer l'efficacité de notre algorithme de détection. ces critères sont définis comme suit :

- *Taux de vrai positif* : il indique le pourcentage de nœuds correctement détectés.
- *Taux de précision de détection* : il est le rapport entre nombre de nœuds correctement détectés (légitimes ou attaquants) et le nombre total de nœuds présents dans le réseau.
- *FP (faux positif)* représente le nombre de nœuds légitimes qui ont été incorrectement déclarés comme attaquants ;

- *FN (faux négatif)* représente le nombre de nœuds attaquants qui ont été incorrectement déclarés comme légitimes ;
- *TP (vrai positif)* représente le nombre de nœuds attaquants qui ont été correctement déclarés comme attaquants ;
- *TN (vrai négatif)* représente le nombre de nœuds légitimes qui ont été correctement déclarés comme légitimes ;
- *Taux de fausse alerte (FAR)* : il désigne le rapport entre le nombre de nœuds incorrectement déclarés (légitimes ou attaquants) et le nombre total de nœuds présents dans le réseau. Il est calculé en fonction du taux de faux positif (FPR) et du taux de faux négatif (FNR) en utilisant la formule [13] ci-après.

$$FNR = \frac{FN}{FN+TP} \tag{III.4}$$

$$FPR = \frac{FP}{FP+TN} \tag{III.5}$$

$$FAR = \frac{FNR+FPR}{2} \tag{III.6}$$

Le taux de vrai positif est calculé en utilisant la formule suivante :

$$Tauxdevraipositif = \frac{TP}{nombretotald'attaquant} \tag{III.7}$$

La précision est calculée en utilisant la formule suivante :

$$precision = \frac{TP+TN}{TP+TN+FN+FP} \tag{III.8}$$

5. Conclusion

Dans ce chapitre, nous avons proposé notre solution pour détecter des attaques par diminution et par augmentation de rang et mitiger leurs impacts. Dans le chapitre qui suit, nous nous intéresserons à l'évaluation des performances de RPL, de l'impact des attaques sur la topologie et à l'évaluation de l'efficacité de détection des attaques de notre solution à travers une étude expérimentale.

Chapitre IV

Etude expérimentale

1. Introduction

Tester notre solution sur des capteurs est coûteux. C'est pourquoi nous avons fait recours au simulateur/émulateur COOJA. Nous avons choisi d'effectuer nos simulations sur le système d'exploitation bien connu pour l'IoT Contiki car il a une implémentation bien testée de RPL appelée ContikiRPL. ContikiRPL fonctionne sur des liaisons sans fil de faible puissance à forte perte. Dans ce chapitre, nous présentons nos outils de simulation puis nous évaluons les performances de RPL sans attaque, lorsqu'il subit des attaques et lorsque notre solution est déployée.

2. Outils de simulation

2.1. Contiki

Contiki est un système d'exploitation léger, portable, et flexible conçu pour les réseaux à ressources limitées. Il a été développé par une équipe de chercheurs suédois en 2004 et implémenté en langage C (ce qui lui permet une meilleure portabilité). Il est composé de noyau, de bibliothèques, d'ordonnanceur et d'un ensemble de processus. Il se charge de la gestion de la mémoire, du processeur et des périphériques d'entrées/sorties. Une configuration Contiki typique consomme 2 kilo-octets de RAM et 40 kilo-octets de ROM [16]. Il utilise un concept qui a pour objectif d'économiser la mémoire appelé Protothread. un Protothread est une nuance

entre le multi-threading et la programmation évènementielle.

Contiki prend en charge deux types de communication :

— Une première couche de communication appelée Rime qui permet un dialogue vers les capteurs voisins ainsi que le routage. Il prend en charge une transmission fiable.

— Et une deuxième appelée uIP (micro-IP) qui est une version miniature de la pile TCP / IP.

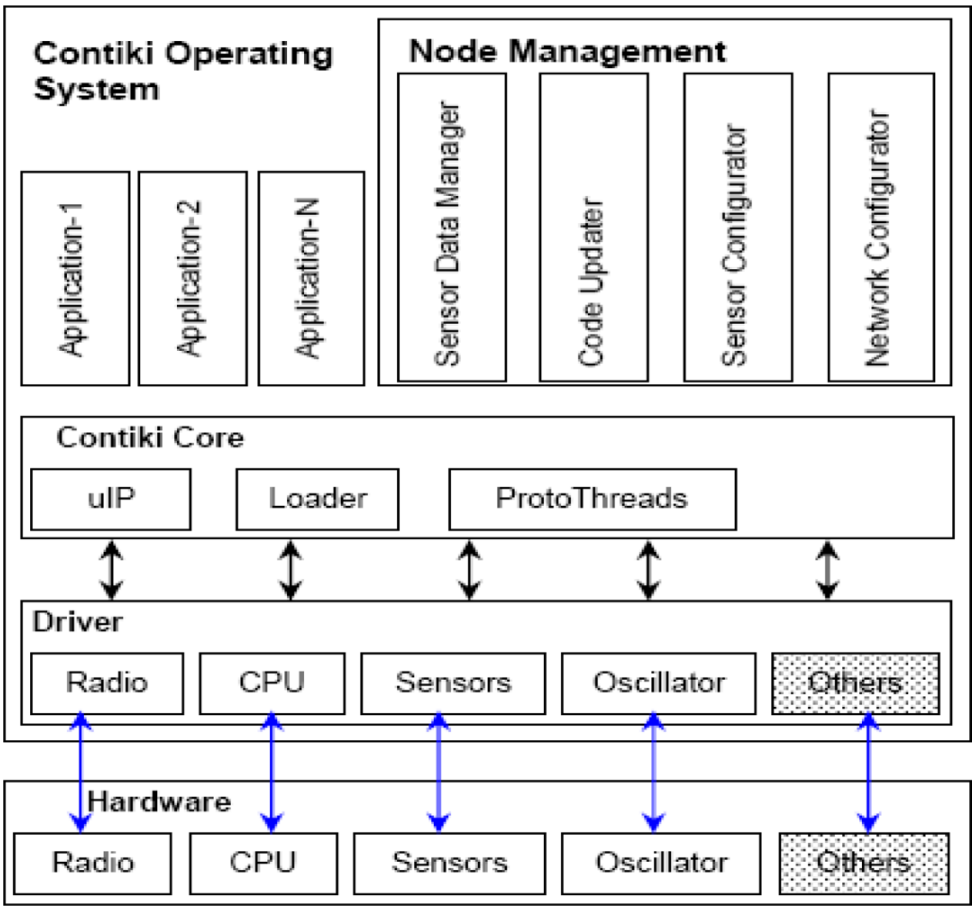

FIGURE IV.1 – Architecture Contiki [16]

2.2. Cooja

Cooja est un simulateur de réseau qui permet l'émulation de plates-formes matérielles réelles. Il simule un réseau de capteurs sans fil en interagissant avec les nœuds.

2.2.1 Les fenetres

Dans le simulateur Cooja, plusieurs fenêtres sont fournies afin de contrôler et de visualiser en temps réel le comportement des nœuds (messages échangés, durée de la simulation,

affichage et l'emplacement géographique) de la simulation en cours. Les plus importantes sont :

- *Network* : elle affiche la disposition physique ou géographique des motes. Elle nous permet aussi de gérer la topologie (changer la position physique des motes par exemple). Dans cette fenêtre les adresses ip, le type de mote et le trafic radio entre les eux peuvent être visualisés en temps réel.
- *Simulation Control* : elle possède les boutons Start, Pause, Step et Reload qui permettent de contrôler la simulation. En effet, elle nous aide à démarrer, recharger, mettre en pause et modifier la vitesse de la simulation en cours.
- *Notes* : elle permet d'ajouter des notes à la simulation en cours.
- *Mote Output* : elle affiche les activités de toutes des motes connectées à la simulation.

Les fenêtres sont illustrées sur la figure IV.2.

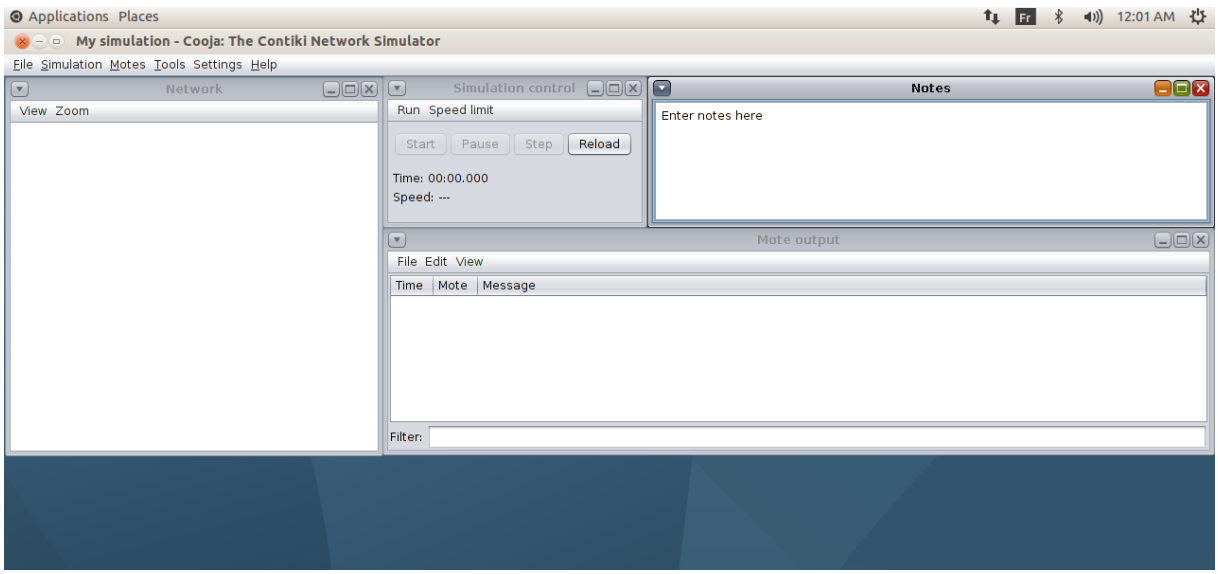

FIGURE IV.2 – Fenêtres Cooja

2.2.2 Mote

Les motes sont des nœuds IoT équipés de capteurs. Ils disposent d'une mémoire et d'une unité de traitement très limitée. Dans le simulateur, nous avons accès à plusieurs motes telsque : Wismote, Z1 Mote, Sky Mote etc. Dans le cadre de notre projet, nous utilisons le *Sky mote*. Il est doté d'un microcontrolleur MSP430, d'une mémoire RAM de 10KB et ROM de 48KB.

2.3. CollectView

CollectView est un outil développé en java, utilisé pour collecter des informations à partir des motes (équipement IoT) et pour leurs envoyer des commandes. Le Sink commence la collecte en envoyant un message aux nœuds du réseau. Ainsi, Chaque nœud envoie sa consommation d'énergie, le taux d'occupation du processeur, les données de son environnement (temperature, humidité et lumière). Ces informations nous permettrons d'effectuer des études statistiques du réseau et de les représentées graphiquement.

3. Paramètres de simulation

Nous simulons, à l'aide de l'outil Cooja et la version 3.0 de Contiki, un réseau LLN composé de plusieurs nœuds contraints. Nous réaliserons plusieurs scénarios de tests afin d'évaluer les performances de notre réseau. Ces scénarios nous permettrons d'effectuer une série de comparaison entre un réseau composé de :

- Nœuds légitimes uniquement ;
- Nœuds légitimes et attaquants ;
- Nœuds légitimes et attaquants dans le quel notre contre-mesure est utilisée.

Nous nous basons sur les critères suivants pour comparer les LLN :

- Énergie : il représente la consommation énergetique moyenne des nœuds ;
- Taux de perte de paquet : il est l'estimation du nombre de paquet perdu dans le réseau ;

Étant donnée que dans la vie réelle des perturbations (interférence et bruit) se produisent dans les canaux de transmission des réseaux sans fil, nous modifions les taux de transmission (TX) et de réception (RX) de paquet avec succès de nos nœuds afin que le réseau simulé reflète le plus un réseau LLN réel composé de nœuds réels.

Nous rappelons que les résultats des tests de performance du LLN obtenus à l'aide de Cooja sont équivalents aux résultats du déploiement d'un LLN dans la vie réelle.

La table IV.1 représente les paramètres de la simulation.

TABLE IV.1 – Table résumant les paramètres de la simulation

Nombre de nœud	10,20,30,40,50
Type de nœud	Sky mote
Surface	300m*300m
RX et TX	90%
Portée Radio des nœuds	50m
Durée	30minutes
Topologie	grille
Métrique de routage	ETX

4. Scénarios d'évaluation

Nous décrivons dans cette section les différents scénerios que nous utilisons pour nos évaluation.

Scénario n °1

Dans ce scénario, nous simulons un réseau LLN dans les conditions normales en variant le nombre de nœuds. Tous les nœuds sont légitimes et implémentent le code par défaut défini par contiki. Ils envoient leurs informations (Utilisation du CPU, Charge restante etc.) au point de collecte qui n'est autre que le sink. Les données collectées par le sink nous permettrons de réaliser une étude statistique dans le but de comparer les performances des différents réseaux.

Scénario n °2

Dans ce scénario, le réseau LLN que nous simulons est formé de nœuds légitimes et attaquants.

Les nœuds malveillants implémentent un code modifié qui leur permettront d'effectuer une attaque de rang dans le réseau. Ils participent à la formation du DODAG sans comportement malicieux. Autrement dit, ils agissent comme des nœuds légitimes (ils envoient des messages DIS afin de solliciter les DIO de leur voisinage). Ensuite ils calculent leur rang (sans aucune falsification) et se positionnent dans le DODAG. Lorsque le réseau converge, Ils changent illégalement leur rang en l'augmentant ou en le diminuant. Dans le cas d'une diminution de rang, ils attirent plus de traffic vers eux. Ainsi nous mesurerons les performances du réseau.

Scénario n °3

Dans ce scénario, les nœuds du réseau LLN du scénario n°2 implémentent notre solution. Nous testerons que les nœuds malveillants sont bien détectés et évaluerons ensuite les performances du réseau.

5. Implémentation des attaques

Nous avons implémenté l'attaque par diminution de rang de manière que l'attaquant diminue son rang mais qu'il ne tire pas trop d'attention. C'est à dire qu'il réduit illégalement son rang de sorte que celui-ci soit supérieur au rang de son parent préféré mais inférieur au rang qu'il doit avoir (rang calculer avec la formule III.1). Ce rang ne permettra pas à l'attaquant d'attirer ses parents et ses frères.

En ce qui concerne l'attaque par augmentantion de rang, l'attaquant augmente son rang de sorte que celui-ci soit supérieur au rang qu'il est censé avoir.

Pour implémenter les attaques, nous avons modifié la fonction *calculate_rank(rpl_parent *p, rpl_rank_t base_rank)* du fichier *core/net/rpl/rpl-mrhof.c*. Cette dernière fonction calcule le rang des nœuds et retourne :

- *Rang_du_parent_préféré+cout_de_chemin+n* dans le cas d'une attaque par augmentation de rang ;
- *Rang_du_parent_préféré+cout_de_chemin-k* dans le cas d'une attaque par diminution de rang. La valeur retournée est supérieure au Rang_du_parent_préféré.

6. Implémentation de la solution

Notre solution est constituée principalement de deux grandes fonctions : la détection d'attaquants et la transmission de leurs informations aux autres nœuds du réseau.

Nous avons commencé par créér une bubliothèque *tools.h* et un fichier *tools.c* dans lequel est codé nos fonctions. Chaque nœud légitime appele la bubliothèque tools.h afin de détecter les attaquants et de prendre des mesures vis-à-vis d'eux.

La fonction *Detecter_attaque()* : Cette fonction recupère dans la table de routage les fils

directs (fils qui sont à 1 saut) du nœud. Puis elle recupère dans la table de voisinage ETX entre le nœud et ses fils directs. Avec ses informations, la fonction estime le rang de tous les fils et vérifie qu'aucun d'eux n'a un comportement malicieux. Autrement dit qu'aucun des nœuds fils n'a un rang supérieur ou inférieur au rang estimé par le nœud. Ceux qui ont un comportement malicieux sont ajoutés à *liste de nœud suspect*. S'ils existaient dans cette liste, ils sont ajoutés à la *liste noire* et leurs informations sont transmises à tous les nœuds du réseau.

La fonction *transmettre d'information()* : Lorsqu'un nœud détecte un attaquant, il transmet les informations de l'attaquant à tous ses voisins qui par la suite retransmettent à leurs voisins et ainsi de suite.

7. Evaluation de RPL

Nous avons simulé plusieurs fois, à l'aide de Cooja, un réseau pendant trente (30) minutes puis nous avons mesuré ses performances et représenté sous forme graphique les resultats. Dans les différentes simulations, nous avons fait varier le nombre de nœud du reseau (10, 20, 30, 40 et 50). Tous les nœuds sont légitimes. La figure IV.3 répresente le taux de perte et la figure IV.4 répresente la consommation électrique moyenne en fonction du nombre de nœud.

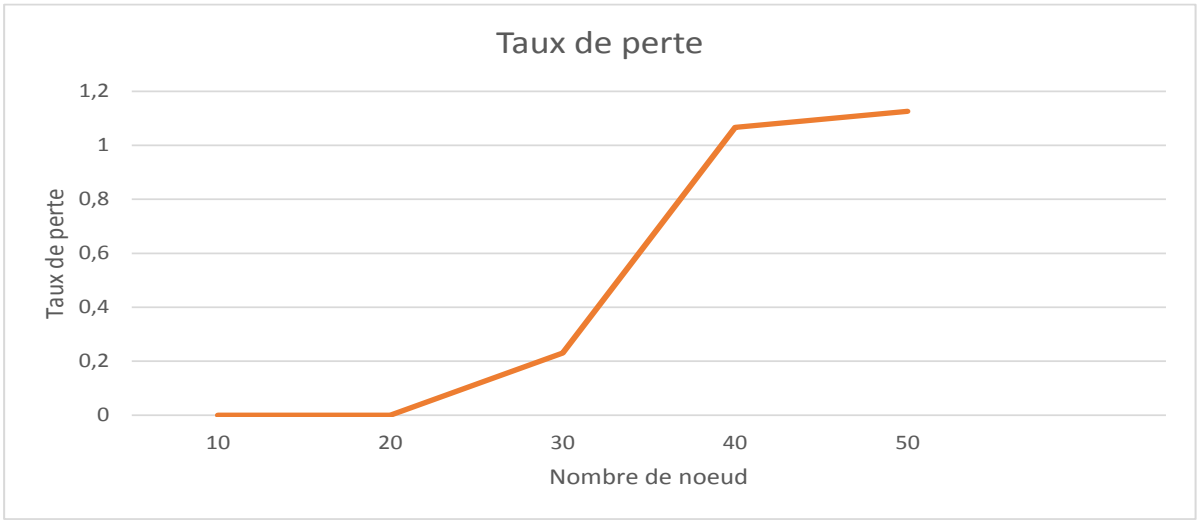

FIGURE IV.3 – Taux de perte dans RPL sans attaques

À partir des resultats de nos simulations, nous déduisons que le taux de perte et la consommation énergetique dans les LLN dépendent en grande partie du nombre de nœud formant le réseaux.

Sur la figure IV.3, le taux de perte croît et sa valeur varie entre 0 et 1,12. Entre 0 et 20 nœuds, le taux de perte est presque nul. C'est à partir de 20 nœuds qu'il croît legèrement. Entre 30 et 50 nœuds, il est fortement croissant.

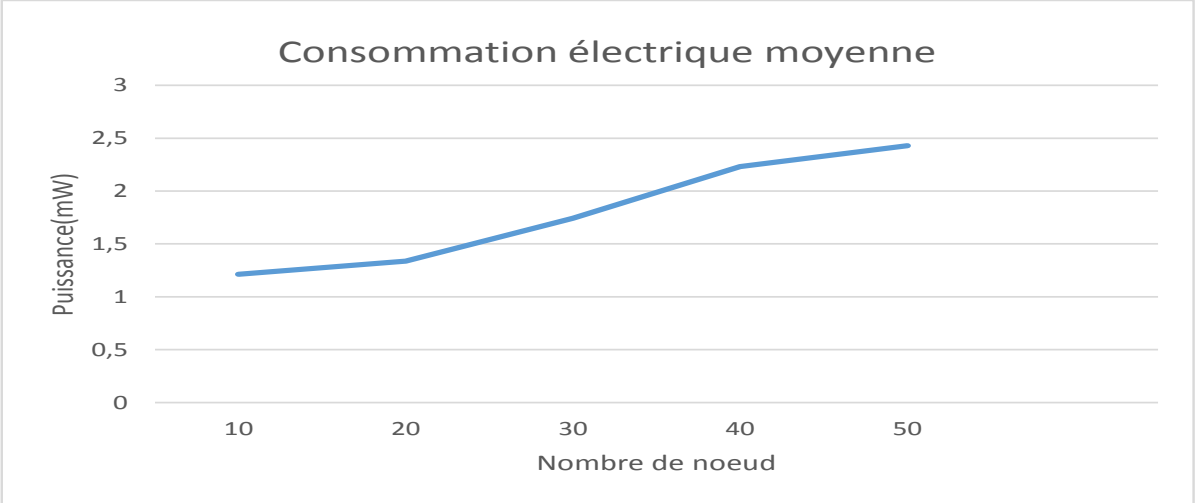

FIGURE IV.4 – Consommation d'énergie dans RPL sans attaques

Ces variations sont expliquées d'une part par la nature des réseaux LLN et d'autre part, par le fait qu' il y'a plusieurs paquets de données dans le réseaux qui doivent être transmis au sink et cela en fontion de la taille du réseaux. En outre, les nœuds qui sont positionnés plus haut dans le DODAG peuvent aussi causer une importante perte car ils relayent tous les paquets vers le sink bien qu'ils ont des ressources limitées. Par conséquent, ils ne pourront plus traiter les paquets entrant au-delà d'un seuil (limite mémoire).

Sur la figure IV.4, nous remarquons que la consommation énergetique augmente avec la taille du réseau. En moyenne, elle varie entre 1,2 et 1,74 pour les réseaux formés de 1 à 30 nœuds. Puis elle croît constamment en fonction du nombre de nœud. En effet, la consommation énergetique d'un nœud dépend de l'utilisation de son cpu et de ses liaisons (transmission et reception de données).

8. Evaluation de l'impact des attaques par diminution de rang (attaque decreased)

Nous avons implémenté l'attaque par diminution de rang puis mesuré son impact dans les réseaux LLN grâce à Cooja. Dans chaque simulation, nous avons fait varier le nombre de nœuds. Le ratio de nœuds attaquants est 10%.

Nous avons la figure IV.5 qui illustre le taux de perte et la figure IV.6 qui répresente la consommation électrique moyenne en fonction du nombre de nœud.

Sur la figure IV.5 nous constatons qu'à partir de 10 nœuds, le taux de perte, dans le cas d'attaque par diminution de rang, croît de façon uniforme jusqu'à 30 nœuds alors que dans RPL

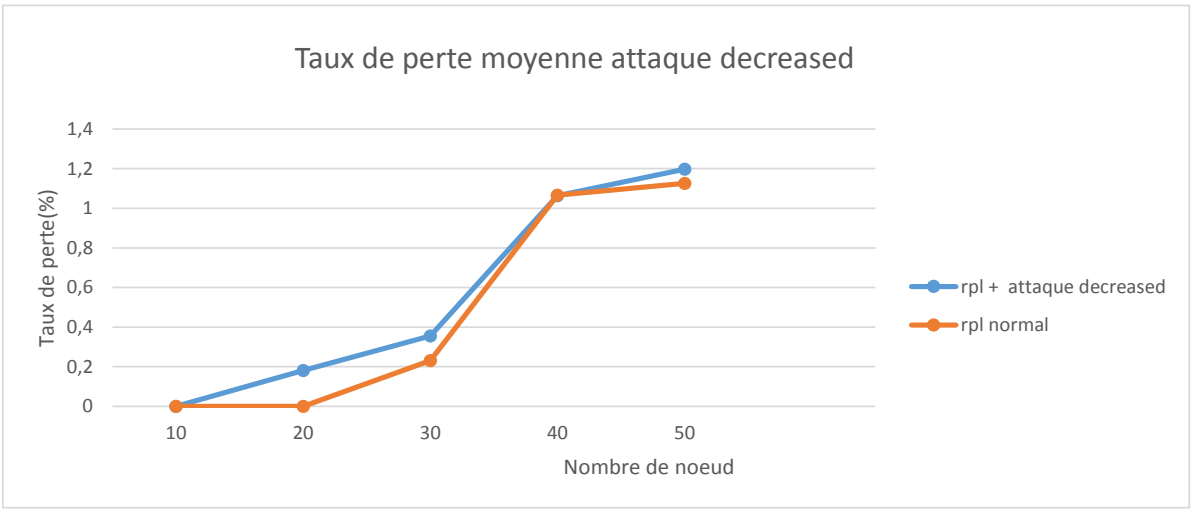

FIGURE IV.5 – Taux de perte dans RPL avec attaques decreased

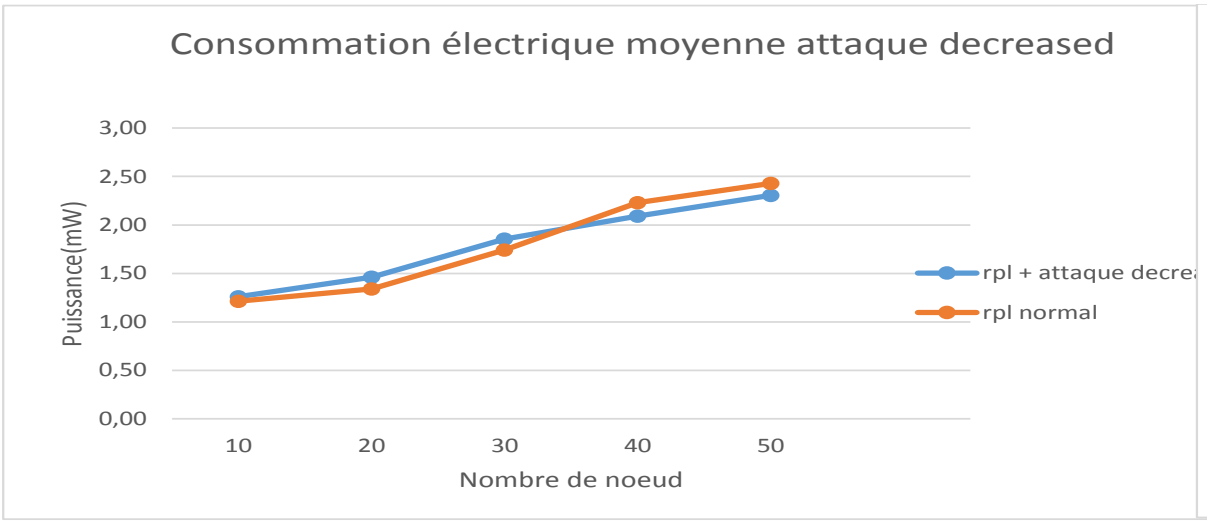

FIGURE IV.6 – Consommation d'énergie dans RPL avec attaques decreased

sans attaque, il était nul jusqu'à 20 nœuds. Entre 30 nœuds et 50, il croît considérablement. En effet après la réalisation de l'attaque, les attaquants ont attiré plus de traffic vers eux. De cela resulte des chemins non optimisés et plus de surcharge dans leur sous-DODAG. Ce qui a finalement causé cette variation du taux de perte.

Sur la figure IV.6, tout comme dans RPL sans attaque, la consommation d'énergie augmente en fonction de la taille du réseau. En outre, nous avons observé de près l'impact de cette attaque. Nous avons vu qu'elle modifie la structure du DODAG. Nous avons capturé le DODAG avant et après l'attaque. La figure IV.7 illustre le changement de la topologie causé par l'attaque par diminution de rang. Nous avons remarqué que les perfomances du reseau se dégradent au fur et à mesure que le nombre d'attaquant augmente. Et que la consommation énergetique des nœuds attaquants a augmenté. Car ils ont attiré plus de trafic vers eux.

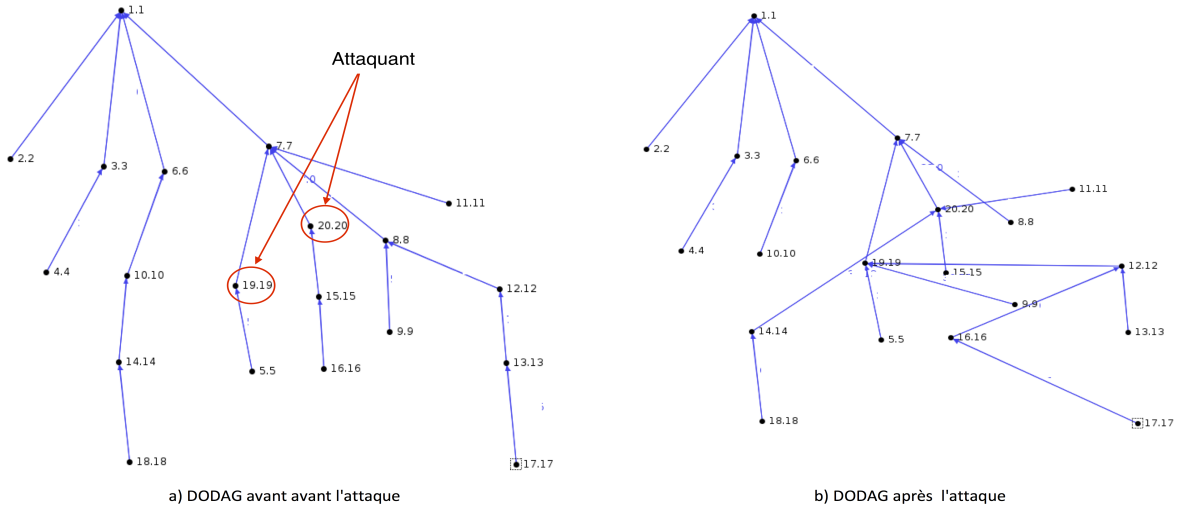

FIGURE IV.7 – Changement de la topologie par l'attaque par diminution de rang

9. Evaluation de l'impact des attaques par augmentation de rang (attaque increased)

Nous avons simulé plusieurs fois des attaques par augmentantion de rang dans un réseau afin de voir son impact. Dans chaque simulation, nous avons fait varier le nombre de nœuds. Le ratio de nœuds attaquants est 10%.

La figure IV.8 illustre le taux de perte et la figure IV.9 répresente la consommation électrique moyenne en fonction du nombre de nœud.

Sur la figure IV.8, le taux de perte croît uniforment à partir de 10 nœuds. Nous remarquons également qu'il a augmenté legèrement par rapport au taux de perte dans RPL sans attaque.

Nous avons remarqué qu'avec l'attaque par augmentantion de rang, les nœuds du sous DODAG de l'attaquant sont victimes d'épuisement de ressource. Pour le prouver, nous avons comparé une attaque increased dans une topologie aléatoirement déployée à une topologie en grille de 20 nœuds. Les résultats sont illustrés sur la figure IV.10. Une très grande différence a été observée. Cela se traduit par le fait que dans la topologie aléatoire il y'a des nœuds qui ont choisi l'attaquant comme parent préféré durant toute la simulation (car ils n'avaient pas de parent meilleure que l'attaquant).

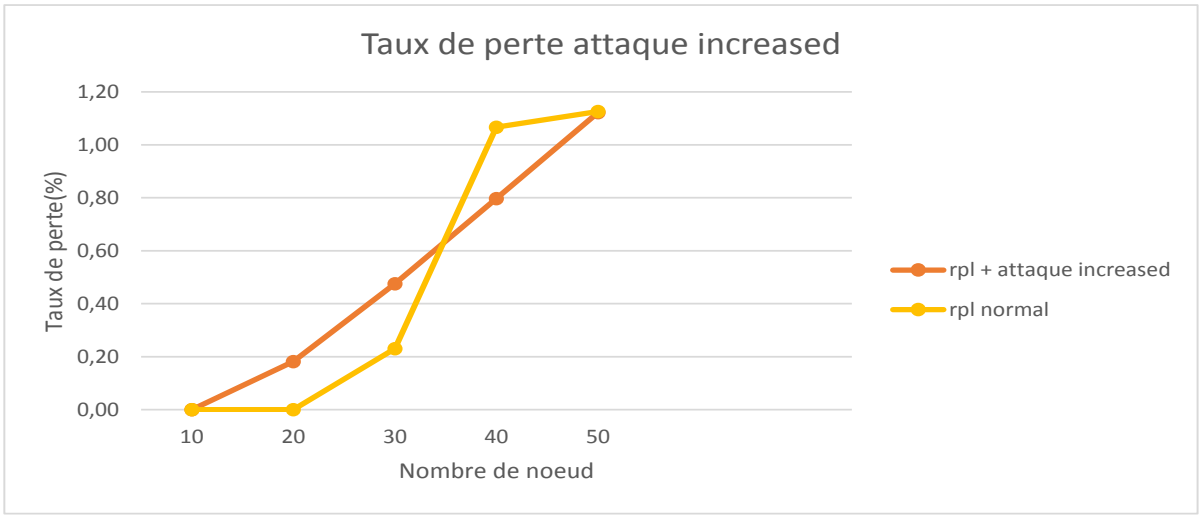

FIGURE IV.8 – Taux de perte dans RPL avec attaques increased

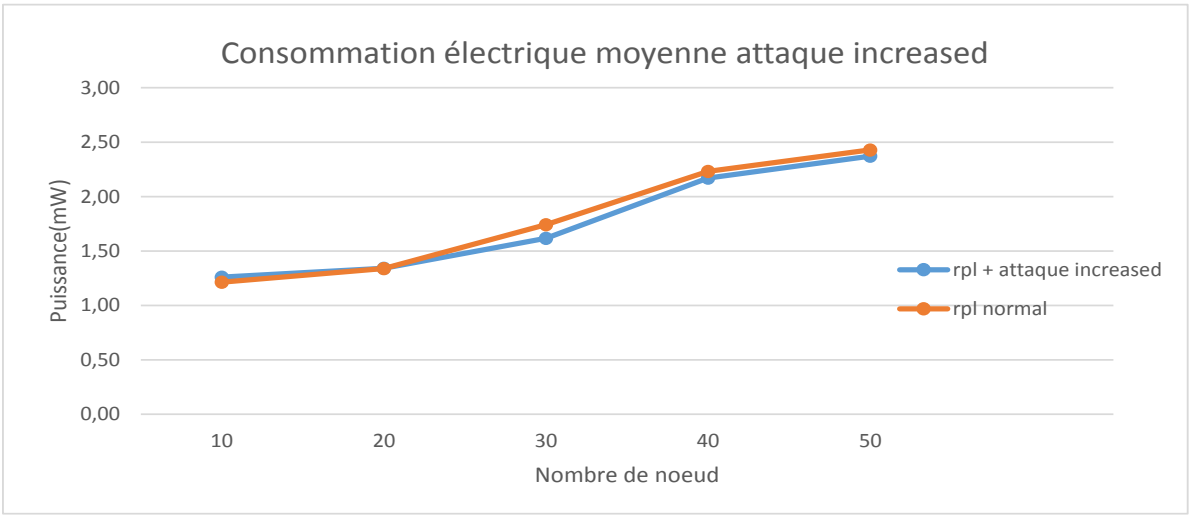

FIGURE IV.9 – Consommation d'énergie dans RPL avec attaques increased

10. Evaluation de l'impact de notre solution

Nous avons simulé les réseaux des deux sections précedentes mais cette fois-ci, les nœuds légitimes utilisent notre solution. Le taux de perte et la consommation énergetique sont répresentés respectivement sur la figure IV.11 et IV.12.

Lorsque les nœuds légitimes utilisent notre solution, sur la figure IV.11, le taux de perte est nul avec les deux attaques jusqu'à 20 nœuds. C'est après 20 nœuds qu'il croît. Lorsque les attaquants ont été détectés, leurs fils qui ont été avertis ont changé de parent préféré. Donc les nœuds attaquants ont été isolés ce qui a causé une amélioration du taux de perte.

Sur la figure IV.12 nous remarquons principalement qu'avec notre solution, la consommation électrique a légerement augmenté par rapport à RPL normal. Cela s'explique par les

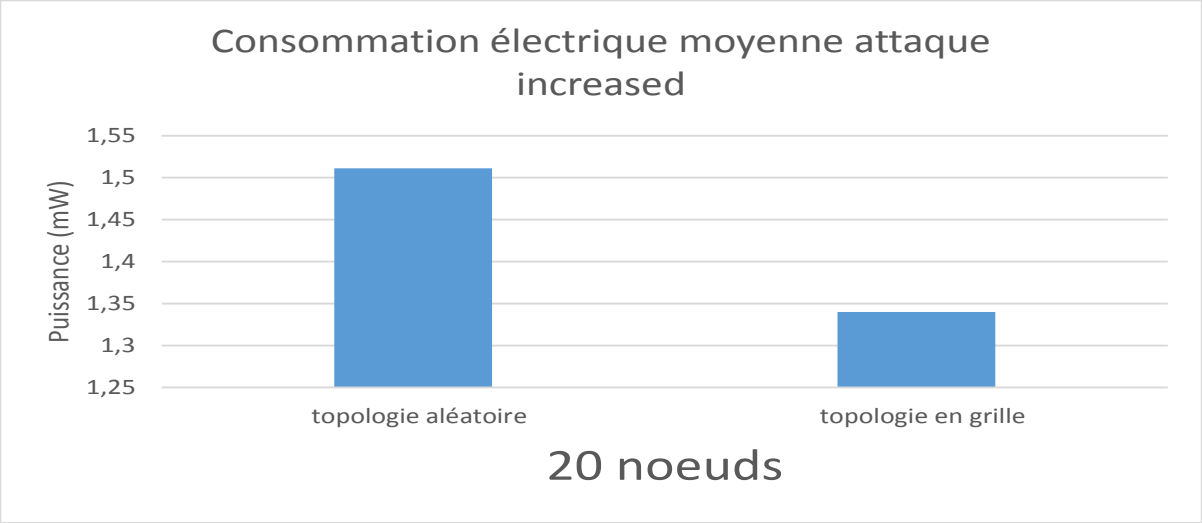

FIGURE IV.10 – Consommation d'énergie attaques decreased topologie en grille et aléatoire

nouveaux traitements qu'effectuent les noeuds pour détecter les attaques.

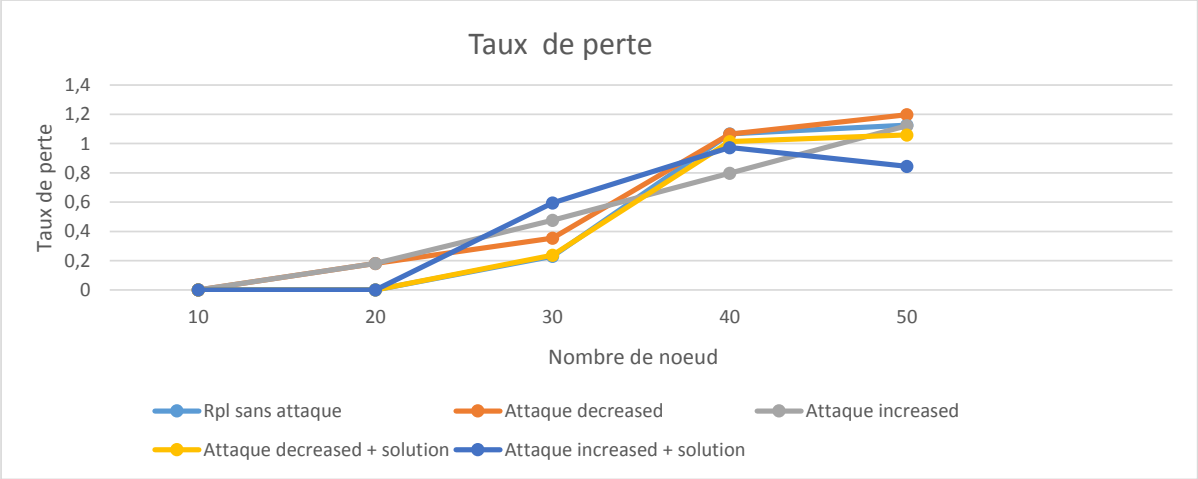

FIGURE IV.11 – Taux de perte dans rpl avec notre solution

11. Evaluation de l'efficacité de notre algorithme de détection

La table IV.2 contient les resultats de l'évaluation de notre algorithme de détection. Les figures IV.13 et IV.14 illustrent respectivement la precision et le taux de fausse alerte de notre solution.

La figure IV.13 et IV.14 illustrent respectivement la précision de détection et le taux de fausse alerte de notre solution.

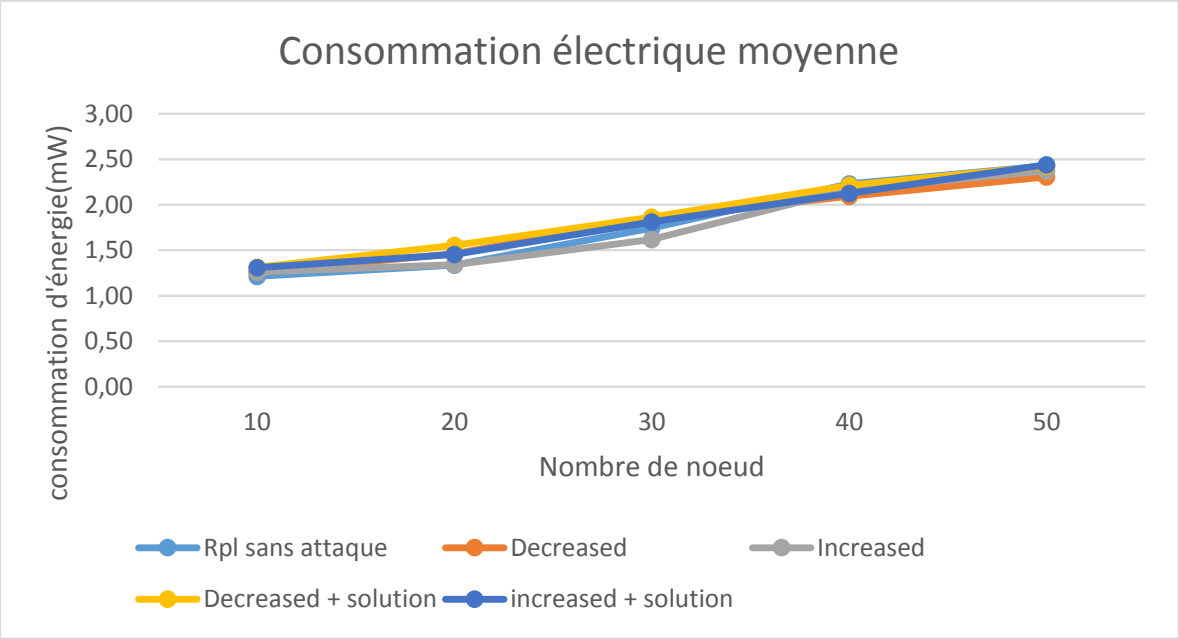

FIGURE IV.12 – Consommation électrique dans rpl avec notre solution

TABLE IV.2 – Table contenant les resultats de simulation

Nombre de nœuds	TP	TN	FP	FN	TFN	TFP	TTP	Precision	FAR
10	1	8	1	0	0	11,11	100	90	5,56
20	1	16	2	0	0	11,11	100	89,47	5,56
30	3	23	4	0	0	14,81	100	86,67	7,41
40	4	25	11	0	0	30,56	100	72,5	15,28
50	5	30	15	0	0	33,33	100	70	16,67

12. Conclusion

Dans ce chapitre, nous avons décrit l'implémentation des attaques et de la solution. Puis nous avons efféctué plusieurs scénarios de simulation. Les resultats obtenus nous ont permis d'évaluer les perfomances de rpl et de notre solution.

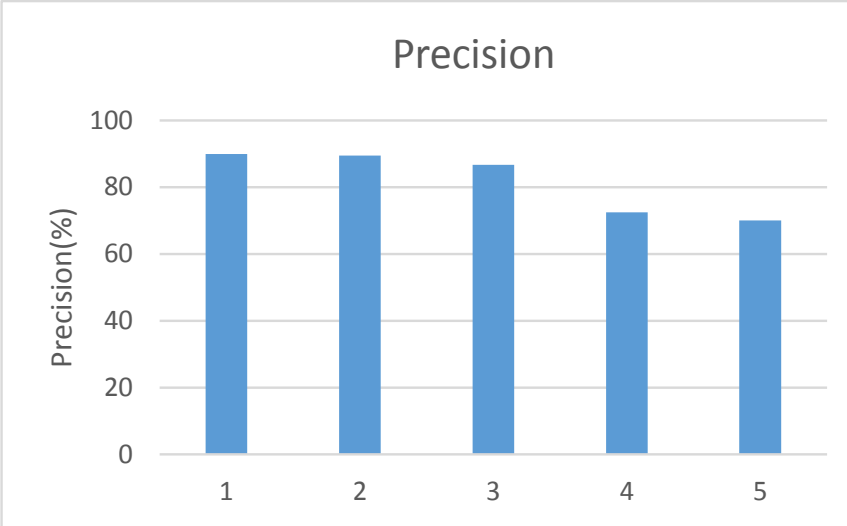

FIGURE IV.13 – Précision de détection d'attaques de la solution

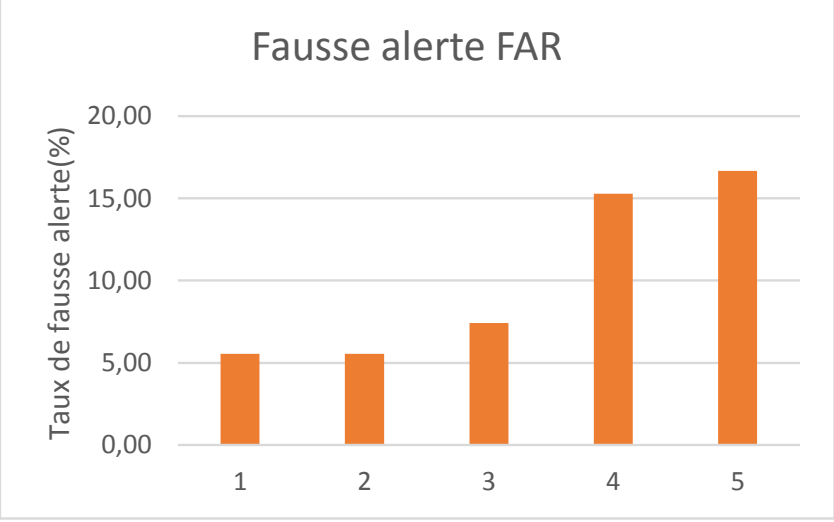

FIGURE IV.14 – Taux de fausse alerte de la solution

Conclusion générale

L' Internet des objets (IoT) a permis aux objets de la vie quotidienne (montres, chaussures, climatiseurs etc.) de se connecter au réseau mondial Internet.

Les réseaux LLN sont des capteurs et des actionneurs aux ressources (puissance de traitement, d'énergie, et mémoire) limitées reliés par des liaisons instables. Ils répresentent une grande partie de l'IoT.

Dans ce mémoire de fin d'étude, nous nous sommes intéressés au routage dans les réseaux LLN et aux risques de sécurité qu'il encourt. L'objectif principal de notre travail, dans un premier temps, était d'étudier et d'analyser les attaques de topologie dont fait l'objet le protocole RPL ainsi que les solutions proposées dans la littérature pour y faire face. Et dans un second temps de proposer et évaluer notre solution afin de détecter et mitiger leurs impacts.

Comme perspectives, nous envisageons que :

- Les résultats de notre travail peuvent être enrichis en éffectuant d'avantages de simulations avec d'avantages de ratio d'attaquants.
- Notre solution peut être améliorée pour détecter les autres types d'attaques que subissent les réseaux LLN.

Bibliographie

Livre

[1] P.J. Benghozi, S. Bureau, F.Massit-Folea : *L'Internet des objets. Quels enjeux pour les Européens ?*, 2008.

[2] R. Dumont : *Cryptographie et Sécurité informatique*, INFO0045-2, 2009

Article de revue

[3] Gaddour, O., Koubâa, A. : *RPL in a nutshell : A survey*, Computer Networks, 56(14), 2012, pp 3163-3178.

[4] Karmakar, S., Sengupta, J., Das Bit, S. : *LEADER : Low Overhead Rank Attack Detection for Securing RPL based IoT*, nov 2020

[5] Raza, S., Wallgren, L., Voigt, T. : *SVELTE : Real-time intrusion detection in the Internet of Things*, Ad hoc networks, 11(8), 2013, pp 2661-2674.

[6] Mayzaud, A., Badonnel, R., Chrisment, I. : *A Taxonomy of Attacks in RPL-based Internet of Things*, International Journal of Network Security, 18(3), May 2016, pp 459-473.

[7] Rghioui, A., Khannous, A., Bouhorma, M. : *Denial-of-Service attacks on 6LoWPAN-RPL networks : Threats and an intrusion detection system proposition*, Journal of Advanced Computer Science & Technology, 3(2), 2014, pp 143.

[8] Almusaylim, Z. A., Alhumam, A., Mansoor, W., Chatterjee, P., Jhanjhi, N. Z. : *Detection and Mitigation of RPL Rank and Version Number Attacks in Smart Internet of Things*, 2020

[9] Le, A., Loo, J., Lasebae, A., Vinel, A., Chen, Y., Chai, M. : *The impact of rank attack on network topology of routing protocol for low-power and lossy networks*, IEEE Sensors Journal, 13(10), oct 2013, pp 3685-3692.

[10] Perrey, H., Landsmann, M., Ugus, O., Schmidt, T. C., Wählisch, M. : *TRAIL : Topology authentication in RPL*, arXiv preprint 2013, arXiv :1312.0984.

[11] Butun, I., Österberg, P., Song, H. : *Security of the Internet of Things : Vulnerabilities, attacks, and countermeasures*, IEEE Communications Surveys & Tutorials, 22(1), oct 2019, pp 616-644.

[12] Silpa Chaitanya, B.Renuka Devi, K.Siva Kumar : *A Novel Method for Secure RPL for Resource Based attacks in Internet of Things* , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN : 2278-3075, Volume-9 Issue-2, Dec 2019

[13] Verma, A., Ranga, V. : *Evaluation of network intrusion detection systems for RPL based 6LoWPAN networks in IoT*, Wireless Personal Communications, 108(3), 2019, pp 1571-1594.

Article d'actes de conférence

[14] Kamble, A., Malemath, V. S., Patil, D. : *Security Attacks and Secure Routing Protocols in RPL-based Internet of Things :Survey*, international Conference on Emerging Trends and Innovation in ICT (ICEI), Pune Institute of Computer Technology, Pune, India, Feb 3-5, 2017, pp 33-39

[15] Dvir, A., Buttyan, L. : *VeRA-version number and rank authentication in RPL*, In 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, Oct 2011 (pp. 709-714), IEEE.

[16] Farooq, M. O., Kunz, T. : *Operating systems for wireless sensor networks : A survey*, Sensors 2011, 11(6), 5900-5930.

RFC

[17] T. Winter, Ed : *IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550*, March 2012 URL : http ://www.ietf.org/rfc/rfc6550

[18] P. Levis : *The Trickle Algorithm. RFC 6206*, March 2011 URL : http ://www.ietf.org/rfc/rfc6206

[19] JP. Vasseur : *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks RFC 6551*, March 2012 URL : http ://www.ietf.org/rfc/rfc6551

[20] O. Gnawali, P. Levis *The Minimum Rank with Hysteresis Objective Function* , September 2012

[21] P. Thubert : *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). RFC 6552*, March 2012 URL : http ://www.ietf.org/rfc/rfc6552s

Thèse

[22] Anthéa, M., : *Monitoring and Security for the RPL-based Internet of Things.* Cryptography and Security [cs.CR]. Université de Lorraine, 2016.

[23] Patrick O. K. : *Configuration dynamique et routage pour l'internet des objets*, Réseaux et télécommunications [cs.NI]. Université de Lorraine, 2017. Français. ffNNT : 2017LORR0241ff. fftel01687704f