UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR
ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

N° d'ordre : 146

**THÈSE DE DOCTORAT UNIQUE**

Présentée par
Dieynaba MALL
pour obtenir le grade de
Docteur de l'Université Cheikh Anta Diop de Dakar

**Mention :** Informatique - Télécommunication

**Spécialité :** Informatique

---

# La Révocation de Clés dans les Réseaux Mobiles Ad Hoc (MANETs) et les Réseaux de Capteurs sans fil (WSNs)

---

Soutenue publiquement le 24 / 02 / 2020 devant le jury composé de :

| | |
|---|---|
| Président : | Hamidou DATHE, Professeur Titulaire, Université Cheikh Anta Diop |
| Rapporteurs : | Mohamed Tayeb LASKRI, Professeur Titulaire, Université Badji Mokhtar |
| | Chérif DIALLO, Maître de Conférences (CAMES), Université Gaston Berger |
| Examinateurs : | Cheikh Ahmadou Bamba GUÈYE, Maître de Conférences (CAMES), Université Cheikh Anta Diop |
| | Ahmad Khoureich KA, Maître Assistant (CAMES), Université Alioune Diop de Bambey |
| Directeur de Thèse : | Karim KONATÉ, Professeur Titulaire, Université Cheikh Anta Diop |

Année universitaire 2019-2020

**Titre : La révocation de clés dans les réseaux mobile ad hoc (MANETs) et les réseaux de capteurs sans fil (WSNs)**

# Résumé

Bien qu'ayant donné aux entreprises, gouvernements et particuliers de nouvelles opportunités, les réseaux traditionnels filaires présentent des inconvénients qui empêchent la réalisation de tout leur potentiel. Les réseaux à configuration automatique sans infrastructure fournissent les moyens de surmonter ces obstacles et les réseaux mobiles ad hoc (MANETs) et les réseaux de capteurs sans fil (WSNs) sont des exemples prometteurs parmi eux. Cependant comme tout réseau de communication, le véritable potentiel de ces deux types de réseaux ad hoc ne peut pas être exploité sans envisager de résoudre de manière adéquate les problèmes de sécurité. Tout comme avec les réseaux traditionnels filaires, avec les MANETs et les WSNs, pour assurer la sécurité des communications, les mécanismes mis en place sont généralement basés sur la cryptographie. Ainsi, pour sécuriser la communication entre deux entités, ces dernières doivent avoir une valeur secrète ou une clé cryptographique. Ceci fait de la gestion des clés (cryptographiques) un service primordial pour la sécurité de n'importe quel système basé sur la communication.

Dans le contexte des MANETs et des WSNs, il existe diverses menaces dont la plus intéressante et la plus importante est probablement la subversion des nœuds. Pour faire face à cette menace qui pourrait aboutir à la compromission de la confidentialité des clés, la compromission de l'authenticité des clés et l'usage non autorisé de clés, le processus de gestion des clés doit veiller à ce que les clés compromises soient annulées (révoquées). Ceci étant, dans les MANETs et les WSNs, la gestion des clés présente deux aspects importants notamment la distribution de clés et la révocation de clés.

La problématique liée à la gestion des clés dans réseaux ad hoc en général, dans les (MANETs) et les (WSNs) en particulier a déjà fait l'objet d'une multitude d'étude et de publications. Alors que dans ce domaine, et dans ce type de réseaux la génération et la distribution de clés a largement été étudiée, la révocation de clés a reçu relativement peu d'attention. Un problème de sécurité vital comme celui-ci ne doit-il pas bénéficier d'une reconnaissance plus appropriée ou doit-il juste être considéré comme un segment partiel de gestion des clés qui peut être négligé ?

De plus en plus les chercheurs se rendent compte de l'importance du domaine et tentent de lui donner la place adéquate en proposant des solutions de révocation de clés. Toutefois malgré l'utilisation de diverses techniques et approches par les concepteurs de protocoles de révocation de clés pour les MANETs et les WSNs, les problèmes non résolus restent multiples. Nous nous proposons alors dans cette thèse de renforcer cette démarche par la fourniture d'une étude comparative exhaustive des différentes approches et solutions de révocation de clés proposées pour les MANETs et les WSNs afin de faire ressortir les forces et les faiblesses de chacune de celles-ci, mettre la lumière sur les défis qui sont restés non résolus et ouvrir des nouveaux axes de recherche. En nous basant sur ces résultats, nous présentons deux solutions SECRET et ECL-EKM, qui visent à améliorer certaines approches et solutions de révocation de clés existantes dans les MANETs et les WSNs respectivement.

**Mots clés :** MANETs, WSNs, Sécurité, Cryptographie, Clé, Certificat, Révocation.

**Title : Key revocation in mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs)**


## Abstract

While giving to compagnies, governments, and individuals new opportunities, traditional wireline networks have drawbacks that impede the realization of their full potential. Automatically configured networks without infrastructure provide the means to overcome these obstacles, and mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) are promising examples among them. However, like any communication network, the true potential of these two types of ad hoc networks can not be exploited without considering adequately addressing security issues. As with traditional wired networks, with MANETs and WSNs, to ensure the security of communications, the mechanisms put in place are generally based on cryptography. Thus, to secure communication between two entities, these must have a secret value or a cryptographic key. This makes the (cryptographic) key management a vital service for the security of any communication-based system.

In the context of MANETs and WSNs, there are various threats, the most interesting and most important of which is probably the subversion of nodes. To address this threat, which could lead to the compromise of the confidentiality of keys, the compromise of the authenticity of keys and the unauthorized use of keys, the key management process must ensure that compromised keys are canceled (revoked). Thus, in MANETs and WSNs, key management has two important aspects including key distribution and key revocation.

The issue of key management in ad hoc networks in general, in MANETs and WSNs in particular has already been the subject of a multitude of studies and publications. While in this area, and in this type of networks the generation and distribution of keys has largely been studied, the revocation of keys has received relatively little attention. Should a vital security issue like this not be given more appropriate recognition, or should it just be considered as a partial segment of key management that can be overlooked?

More and more researchers are realizing the importance of the domain and trying to give it the proper place by offering key revocation solutions. However, despite the use of various techniques and approaches by the designers of key revocation protocols for MANETs and WSNs, unresolved issues remain multiple. In this thesis, we propose to reinforce this approach by providing an exhaustive comparative study of the different approaches and key revocation solutions proposed for MANETs and WSNs, in order to highlight the strengths and weaknesses of each of them, shed light on the challenges who remained unresolved and open up new lines of research. Based on these results, we present two solutions SECRET and ECL-EKM, which aim to improve some existing key revocation approaches and solutions in MANETs and WSNs respectively.



**Keywords:** MANETs, WSNs, Security, Cryptography, Key, Certificate, Revocation.

# Table des matières

**INTRODUCTION GENERALE** ..... **1**
**Première Partie :** ..... **7**
**REVUE DE LA** ..... **7**
**LITTERATURE** ..... **7**
**Chapitre 1 CONCEPTS GENERAUX DES MANETs ET WSNs** ..... **6**
**1.1 Introduction aux réseaux sans fil** ..... **6**
1.1.1 Les types de réseaux sans fil ..... **6**
1.1.1.1 Les réseaux sans fil avec infrastructure fixe (SBS ou Basic Service Set) ..... 6
1.1.1.2 Les réseaux mobiles ad hoc (ISBS ou Independent Basic Service Set) ..... 7
1.1.2 Les gammes de technologies sans fil ..... **8**
1.1.2.1 Les Wireless Personal Area Network (IEEE 802.15) ..... 9
1.1.2.2 Les Wireless Local Area Network (IEEE 802.11) ..... 10
1.1.2.3 Les Wireless Metropolitan Area Network (IEEE 802.16) ..... 10
**1.2 Les Réseaux Mobiles Ad Hoc (MANETs)** ..... **11**
1.2.1 Définition d'un réseau mobile Ad hoc ..... **11**
1.2.2 Caractéristiques des réseaux mobiles Ad hoc ..... **12**
1.2.3 Domaines d'applications des MANETs ..... **13**
1.2.4 Les Problèmes et les contraintes des MANETs ..... **14**
**1.3 Les réseaux de capteurs sans fil (WSNs)** ..... **15**
1.3.1 Définition d'un réseau de capteurs sans-fil ..... **15**
1.3.2 Objectifs de base des WSNs ..... **15**
1.3.3 Types de WSNs ..... **16**
1.3.4 Architecture de base d'un capteur ..... **16**
1.3.5 Structure d'un réseau de capteur sans fil ..... **17**
1.3.6 Facteurs et Contraintes Conceptuelles du Réseau et des Protocoles ..... **18**
1.3.7 MANETs versus WSNs ..... **20**
**Conclusion** ..... **21**
**Chapitre 2 SECURITE ET GESTION DE CLES DANS LES MANETs ET LES WSNs** ..... **22**
**2.1. Sécurité des MANETs et des WSNs** ..... **22**
2.1.1 Les Attaques de Sécurité ..... **24**
2.1.1.1 Classification basée sur la légitimité ..... 24
2.1.1.2 Classification basée sur l'interaction ..... 24
2.1.1.3 Classification d'attaques par pile de protocole réseau ..... 25
2.1.2 Les Services de Sécurité ..... **27**

2.1.3 Mécanismes de sécurité ... **29**
2.1.3.1 Cryptographie symétrique ... 30
2.1.3.2 Cryptographie asymétrique ... 31
2.1.3.3 Fonctions de hachage cryptographiques ... 32
2.1.3.4 Message Authentication Code ... 32
2.1.3.5 Signature numérique ... 33
2.1.3.6 Certificat électronique ... 34
**2.2. La Gestion des clés ... 34**
2.2.1 Trusted Third Parties (TTPs) ... **35**
2.2.2 Public Key Infrastructure (PKI) ... **36**
**2.3. Gestion des clés dans les MANETs et les WSNs ... 37**
2.3.1 Contraintes de conception ... **37**
2.3.2 Les Approches de Gestion des clés dans les MANETs et les WSNs ... **39**
**2.4. La Révocation de Clés dans les MANETs et les WSNs ... 40**
2.4.1 Définition de la Révocation de clés ... **40**
2.4.2 Importance et Négligence de la Révocation ... **41**
2.4.3 Contraintes de conception de solutions de révocation de clés ... **42**
**Conclusion ... 43**
**Chapitre 3 LA REVOCATION DE CLES DANS LES MANETS ET LES WSNS ... 44**
**3.1 Classification des Protocoles de Révocation de Clés dans les MANETs et les WSNs ... 44**
**3.2 Les protocoles de Révocation de clés dans les MANETs ... 45**
3.2.1 Les protocoles Distribués ... **45**
3.2.2 Les protocoles Décentralisés ... **51**
3.2.3 Les protocoles Hybrides ... **53**
**3.3 Les protocoles de Révocation de clés dans les WSNs ... 55**
3.3.1 Protocoles Centralisés ... **55**
3.3.2 Protocoles Distribués ... **60**
3.3.3 Protocoles Hybrides ... **64**
**Conclusion ... 68**
**Deuxième Partie : CONTRIBUTIONS ... 70**
**Chapitre 4 ANALYSE DES PROTOCOLES DE RÉVOCATION DE CLÉS DANS LES MANETS ... 70**
**4.1 Critères d'évaluation des protocoles ... 70**
4.1.1 Sécurité ... **70**
4.1.2 Efficacité ... **71**
4.1.3 Applicabilité ... **72**

**4.2 Evaluation de la sécurité des protocoles de révocation de clés dans les MANETs ..... 72**
4.2.1 Les protocoles Distribués ..... **72**
4.2.1.1 Attaques de type Man-in-the-middle ..... 72
4.2.1.2 Attaque par exhaustion de batterie ..... 73
4.2.1.3 Attaque par fausse accusation ou Bad Mouth Attack (BMA) ..... 74
4.2.1.4 Attaque sybil, attaques par coalition de nœuds ..... 74
4.2.1.5 Attaques d'itinérance ..... 76
4.2.2 Les protocoles décentralisés ..... **77**
4.2.3 Les protocoles Hybrides ..... **78**
4.2.3.1 Attaques de type Man-in-the-middle, attaques sybil et attaques par réplication ..... 78
4.2.3.2 Attaques par exhaustion de batterie ..... 79
4.2.3.3 Attaque par fausse accusation ou *Bad Mouth Attack (BMA)* ..... 79
4.2.3.4 Attaques par coalition de nœuds ..... 80
**4.3 Evaluation de l'efficacité des protocoles de révocation de clés dans les MANETs ..... 81**
4.3.1 Protocoles Distribués ..... **81**
4.3.1.1 Charge de stockage ..... 81
4.3.1.2 Charge de communication ..... 83
4.3.1.3 Charge de calcul ..... 84
4.3.1.4 Délai ou latence de la révocation ..... 88
4.3.2 Protocoles Décentralisés ..... **90**
4.3.2.1 Charge de stockage ..... 90
4.3.2.2 Charge de communication ..... 91
4.3.2.3 Charge de calcul ..... 92
4.3.2.4 Délai ou latence de la révocation ..... 92
4.3.3 Protocoles Hybrides ..... **93**
4.3.3.1 Charge de stockage ..... 93
4.3.3.2 Charge de communication ..... 94
4.3.3.3 Charge de calcul ..... 94
4.3.3.4 Délai ou latence de la révocation ..... 95
**4.4 Evaluation de l'Applicabilité des protocoles de révocation de clés dans les MANETs . 96**
4.4.1 Type de réseau ..... **96**
4.4.2 Taille du réseau ..... **96**
4.4.3 Mobilité des nœuds ..... **97**
**4.5 Discussions et Conclusion ..... 101**
**Chapitre 5 ANALYSE DES PROTOCOLES DE RÉVOCATION DE CLÉS DANS LES WSNs ..... 104**

**5.1 Evaluation de la sécurité des protocoles de révocation de clés dans les WSNs ..... 104**
5.1.1 Protocoles Centralisés ..... **104**
5.1.1.1 Attaques de type Man-in-the-middle (MiM), Attaque DoS (exhaustion de batterie) 104
5.1.1.2 Attaque sybil, attaque par réplication de nœuds ..... 105
5.1.1.3 Unique Point de défaillance ..... 106
5.1.2 Protocoles Distribués ..... **106**
5.1.2.1 Attaques de type man-in-the-middle et attaques par exhaustion de batterie ..... 106
5.1.2.2 Attaques par Déni de service (DoS) ..... 107
5.1.2.3 Attaques sybil, attaques par réplication de nœuds ..... 107
5.1.2.4 Attaques via les clés de chemin ..... 107
5.1.2.5 Attaques par fausse accusation, attaques par coalition de nœuds ..... 108
5.1.3 Protocoles Hybrides ..... **108**
**5.2 Evaluation de l'Efficacité des protocoles de révocation de clés dans les WSNs ..... 111**
5.2.1 Protocoles Centralisés ..... **111**
5.2.1.1 Charge de stockage ..... 111
5.2.1.2 Charge de communication ..... 111
5.2.1.3 Charge de calcul ..... 113
5.2.1.4 Délai ou latence de la révocation ..... 114
5.2.2 Protocoles Distribués ..... **116**
5.2.2.1 Charge de stockage ..... 117
5.2.2.2 Charge de communication ..... 118
5.2.2.3 Charge de calcul ..... 119
5.2.2.4 Délai ou latence de la révocation ..... 121
5.2.3 Protocoles Hybrides ..... **123**
5.2.3.1 Charge de stockage ..... 123
5.2.3.2 Charge de communication ..... 124
5.2.3.3 Charge de calcul ..... 126
5.2.3.4 Délai ou latence de la révocation ..... 127
**5.3 Evaluation de l'Applicabilité des protocoles de révocation de clés dans les WSNs ..... 130**
5.3.1 Taille de réseau ..... **130**
5.3.2 Mobilité des nœuds ..... **130**
**5.4 Discussions et Conclusion ..... 135**
**Chapitre 6 PROPOSITION DE SOLUTION POUR LA REVOCATION DE CLÉS DANS LES MANETs ..... 137**
**6.1 Contexte et Motivation ..... 137**

**6.2 Fondements de SECRET ..... 139**
6.2.1 Choix des Protocoles de Référence ..... **139**
6.2.2 Description du protocole Hoeper et Gong (2009) ..... **140**
6.2.3 Description du protocole d'authentification HEAP ..... **142**
**6.3 Adaptation de la solution de Hoeper aux systèmes basés sur PKI ..... 144**
6.3.1 Description de SECRET ..... **144**
**6.4 Analyse de Sécurité et de Performance ..... 150**
6.4.1 Analyse de sécurité ..... **150**
6.4.2 Analyse de Performance ..... **153**
**Conclusion ..... 156**
**Chapitre 7 SOLUTION POUR L'OPTIMISATION DU PROTOCOLE DE GESTION DE CLÉS "CL-EKM" ..... 157**
**7.1 Contexte et Motivation ..... 157**
**7.2 Fondements de notre Proposition ..... 158**
7.2.1 Choix du Protocole Seo et al. (2015) [29] ..... **158**
7.2.2 Choix du Protocole d'authentification ..... **160**
7.2.3 Vue d'ensemble de la solution dans [132] ..... **161**
**7.3 Description de ECL-EKM ..... 162**
7.3.1 Vue d'ensemble de ECL-EKM ..... **162**
7.3.2 Détails du Mécanisme d'authentification proposé par ECL-EKM ..... **162**
7.3.2.1 Génération de blocs étendus ..... 162
7.3.2.2 Diffusion de blocs étendus ..... 164
7.3.2.3 Vérification des blocs étendus ..... 165
**7.4 Analyse de Performance et Résultats ..... 165**
7.4.1 Consommation Energétique ..... **165**
7.4.1.1 Evaluation de la consommation Energétique ..... 165
7.4.2 Délai de Transfert ..... 171
7.4.2.1 Evaluation du Délai ..... 171
**Conclusion ..... 175**
**CONCLUSION GENERALE ..... 176**
**BIBLIOGRAPHIE ..... 182**

# DEDICACE

*A la mémoire de mon cher oncle Lamine Sarr.*

*A mes très chers parents, nulle dédicace n'est susceptible de vous exprimer ma profonde affection, mon immense gratitude pour tous les sacrifices que vous avez consacrés pour moi.*

*A mes petites sœurs Gaysiry et Rokhaya que j'aime tellement*

*A tous ceux qui m'aiment*

# Remerciements

Avant tout, je remercie ALLAH le tout puissant, qui m'a donné la vie et la force pour accomplir cette tâche, qui au début paraissait une mission difficile.

Je tiens à adresser mes plus chaleureux remerciements à Mon professeur et encadreur, Monsieur Karim Konaté et lui exprimer toute ma reconnaissance pour son encadrement, ses conseils, son soutien constant ainsi que pour sa confiance et sa patience. J'ai eu l'honneur et le plaisir de travailler sous sa direction pendant mon stage de DEA et ma thèse de doctorat et j'espère pouvoir continuer à travailler avec lui dans le futur.

Je suis très reconnaissante envers Monsieur Al Sakib-Khan Pathan, mon co-encadrant, pour sa disponibilité malgré la distance qui nous sépare, son ouverture d'esprit, sa générosité et les nombreuses discussions fructueuses qui ont animé ces années de thèse.

Je remercie le Président et les membres du jury notamment :

- Monsieur Hamidou Dathe Professeur Titulaire, Président du jury ;
- Monsieur Mohamed Tayeb Laskri, Professeur Titulaire, Rapporteur ;
- Monsieur Chérif Diallo, Maître de Conférences (CAMES), Rapporteur ;
- Monsieur Cheikh Ahmadou Bamba Guèye, Maître de Conférences (CAMES), Examinateur ;
- Monsieur Ahmad Ka Khoureich, Maître Assistant (CAMES), Examinateur ;
- Monsieur Karim Konaté, Professeur Titulaire, Directeur de thèse

pour avoir relu et évalué mon travail ainsi que pour leurs questions, critiques et suggestions lors de la soutenance.

Je remercie également Monsieur Samuel Pierre pour l'invitation et le soutien qu'il m'a accordé pour un stage de recherche au laboratoire de recherche LARIM de l'école polytechnique de Montréal.

Mes remerciements vont aussi à l'endroit de mes professeurs : Monsieur Mbaye Sène qui m'a beaucoup encouragée lorsque j'ai choisi de poursuivre dans le domaine de la recherche, Monsieur Abdourahmane Raimy, Monsieur Ibrahima Niang, Monsieur Samba N'diaye qui m'ont formée.

Mes vifs remerciements vont à l'ensemble des enseignants de la section Informatique qui m'ont encadrée durant ma formation, mes amis Abdou Rahim Gaye, Fatoumata Baldé, mes camarades de promotion Youssou Kassé, Mamadou Thiongane et Modou Guéye.

J'adresse bien évidemment mes remerciements à ma famille, mes parents, mes frères et sœurs qui m'ont accompagnée dans toutes les entreprises de ma vie.

Je remercie tous ceux qui, de près ou de loin, m'ont apporté leur soutien pour la réussite de ce travail de thèse.

*Merci à tous ceux que j'aime et que je n'ai pas cité*

# Liste des illustrations

Figure 1. 1 : Structure d'un réseau sans fil en mode avec infrastructure ..... 7
Figure 1. 2 : Structure d'un réseau sans fil en mode ad hoc ..... 8
Figure 1. 3 : Différentes catégories de réseaux sans fil [33] ..... 9
Figure 1. 4 : Topologie d'un réseau mobile ad hoc ..... 11
Figure 1. 5 : Architecture de base d'un capteur [44]. ..... 17
Figure 1. 6 : Structure d'un réseau de capteurs sans-fil. ..... 18

Figure 2. 1 : Chiffrement symétrique ..... 30
Figure 2. 2 : Chiffrement asymétrique ..... 31
Figure 2. 3 : Utilisation des fonctions de hachage cryptographique ..... 32
Figure 2. 4 : Utilisation des fonctions de hachage à clé symétrique ..... 33
Figure 2. 5 : Signature numérique ..... 33
Figure 2. 6 : Principaux composants d'une PKI. ..... 37
Figure 2. 7 : Contraintes de conception de solutions de gestion de clés ..... 38
Figure 2. 8 : Classification des systèmes de gestion de clés ..... 40

Figure 3. 1 : Classification des protocoles de révocation de clés dans les MANETs et les WSNs ..... 45
Figure 3. 2 : Architecture de Guo et al. (2014) [83] ..... 51
Figure 3. 3 : Arbre de révocation ..... 56
Figure 3. 4 : Structuration du réseau en cluster ..... 59
Figure 3. 5 : Exemple illustratif de vérification de vote. ..... 63
Figure 3. 6 : Découpage du réseau en regions basiques et non basiques. ..... 65
Figure 3. 7 : Structure en clusters ..... 67

Figure 6. 1 : Le nœud $i$ doit transmettre un paquet à ses voisins $j1$ à $j5$. ..... 142
Figure 6. 2 : Description du fonctionnement de notre solution ..... 146
Figure 6. 3 : Organigramme de l'algorithme N*eighborhood Watch* ..... 147
Figure 6. 4 : Organigramme de l'algorithme *Propagate* ..... 148
Figure 6. 5 : Organigramme de l'Algorithme *Update CRL* ..... 150

Figure 7. 1 : Un exemple de génération de blocs étendus ..... 164
Figure 7. 2 : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de nœuds $N$) ..... 168
Figure 7. 3 : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de sauts $r$ à parcourir) ..... 169
Figure 7. 4 : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de nœuds $N$ et nombre de sauts $r$ à parcourir) ..... 169
Figure 7. 5 : Evolution de la consommation énergétique en fonction de la variation du nombre de messages $m$ de $M$ ..... 170
Figure 7. 6 : Délai de transmission en fonction de la variation de la taille du réseau (nombre de nœuds $N$) ..... 173

Figure 7. 7 : Délai de transmission en fonction de la variation de la taille du réseau (Nombre de sauts à parcourir $r$) ..... 173
Figure 7. 8 : Délai de transmission en fonction de la variation de la taille du réseau ..... 173
Figure 7. 9 : Délai de transmission en fonction de la variation du nombre de messages $m$ de $M$ ..... 174

# Liste des Tableaux

Tableau 1. 1 : Tableau récapitulatif des normes sans fil ..... 11
Tableau 1. 2 : Les différentes technologies des nœuds capteurs ..... 20
Tableau 1. 3 : MANETs versus WSNs..... 21

Tableau 2. 1 : Type d'attaque basée sur l'interaction [53] ..... 25
Tableau 2. 2 : Attaque au niveau de chaque couche [54]. ..... 25

Tableau 4. 1 : Protocoles Distribués – Résumé de l'analyse de sécurité..... 76
Tableau 4. 2 : Protocoles Décentralisés – Résumé de l'analyse de sécurité..... 78
Tableau 4. 3 : Protocoles Hybrides – Résumé de l'analyse de sécurité ..... 80
Tableau 4. 4 : Charge de stockage des Protocoles Distribués – Résumé de l'analyse ..... 82
Tableau 4. 5 : Charge de communication des Protocoles Distribués – Résumé de l'analyse ..... 84
Tableau 4. 6 : Charge de calcul des Protocoles Distribués – Résumé de l'analyse..... 87
Tableau 4. 7 : Délai de révocation des Protocoles Distribués – Résumé de l'analyse ..... 90
Tableau 4. 8 : Charge de stockage des Protocoles Décentralisés – Résumé de l'analyse ..... 91
Tableau 4. 9 : Charge de communication des Protocoles Décentralisés – Résumé de l'analyse ..... 91
Tableau 4. 10 : Charge de calcul des Protocoles Décentralisés – Résumé de l'analyse..... 92
Tableau 4. 11 : Délai de révocation des Protocoles Décentralisés – Résumé de l'analyse ..... 93
Tableau 4. 12 : Charge de stockage des Protocoles Hybrides – Résumé de l'analyse ..... 93
Tableau 4. 13 : Charge de communication des Protocoles Hybrides – Résumé de l'analyse ..... 94
Tableau 4. 14 : Charge de calcul des Protocoles Hybrides – Résumé de l'analyse ..... 95
Tableau 4. 15 : Délai de révocation des Protocoles Hybrides – Résumé de l'analyse ..... 95
Tableau 4. 16 : Applicabilité des Protocoles - Résumé de l'analyse..... 98
Tableau 4. 17 : Avantages et Inconvénients des Protocoles Distribués ..... 99
Tableau 4. 18 : Avantages et Inconvénients des Protocoles Décentralisés ..... 101
Tableau 4. 19 : Avantages et Inconvénients des Protocoles Hybrides ..... 101

Tableau 5. 1 : Protocoles Centralisés – Résumé de l'analyse de sécurité ..... 105
Tableau 5. 2 : Protocoles Distribués – Résumé de l'analyse de sécurité..... 108
Tableau 5. 3 : Protocoles Hybrides – Résumé de l'analyse de sécurité ..... 110
Tableau 5. 4 : Charge de stockage des Protocoles Centralisés – Résumé de l'analyse..... 111
Tableau 5. 5 : Charge de communication des Protocoles Centralisés – Résumé de l'analyse ..... 112
Tableau 5. 6 : Charge de calcul des Protocoles Centralisés – Résumé de l'analyse ..... 114
Tableau 5. 7 : Délai de révocation des Protocoles Centralisés – Résumé de l'analyse..... 116
Tableau 5. 8 : Charge de stockage des Protocoles Distribués – Résumé de l'analyse ..... 117
Tableau 5. 9 : Charge de communication des Protocoles Distribués – Résumé de l'analyse ..... 118
Tableau 5. 10 : Charge de calcul des Protocoles Distribués – Résumé de l'analyse..... 120
Tableau 5. 11 : Délai de révocation des Protocoles Distribués – Résumé de l'analyse ..... 122
Tableau 5. 12 : Charge de stockage des Protocoles Hybrides – Résumé de l'analyse ..... 124
Tableau 5. 13 : Charge de communication des Protocoles Hybrides – Résumé de l'analyse ..... 125
Tableau 5. 14 : Charge de calcul des Protocoles Hybrides – Résumé de l'analyse ..... 127

Tableau 5. 15 : Délai de révocation des Protocoles Hybrides – Résumé de l'analyse ..... 129
Tableau 5. 16 : Applicabilité des Protocoles – Résumé de l'analyse ..... 131
Tableau 5. 17 : Avantages et Inconvénients des Protocoles Centralisés ..... 132
Tableau 5. 18 : Avantages et Inconvénients des Protocoles Distribués ..... 133
Tableau 5. 19 : Avantages et Inconvénients des Protocoles Hybrides ..... 134

Tableau 6. 1 : Sécurité et Efficacité des Protocoles Distribués – Synthèse de l'analyse ..... 139
Tableau 6. 2 : Symboles et notations utilisés ..... 144
Tableau 6. 3 : Comparaison de Performances ..... 155

Tableau 7. 1 : Performance des Protocoles Centralisés – Synthèse de l'analyse ..... 157
Tableau 7. 2 : Liste de Notations ..... 161
Tableau 7. 3 : Paramètres d'entrée (Energie) ..... 167
Tableau 7. 4 : Autres paramètres ..... 168
Tableau 7. 5 : Paramètres d'entrée (Délai) ..... 172

# INTRODUCTION GENERALE

Une recherche approfondie en réseau a créé de nouvelles opportunités pour les entreprises, les gouvernements et les particuliers et a amélioré la productivité globale. Cependant, un réseau traditionnel présente plusieurs inconvénients, qui nous empêchent de réaliser tout son potentiel. L'un des problèmes est qu'il faut une infrastructure établie pour pouvoir fournir des services. L'installation et la maintenance d'un tel réseau peuvent être coûteuses, prendre du temps et être irréalisables dans des régions (très éloignées, difficiles d'accès). Les réseaux à configuration automatique, sans infrastructure, fournissent les moyens de surmonter ces obstacles et les réseaux ad hoc et les réseaux de capteurs spécialisés en sont des exemples prometteurs.

Les réseaux ad hoc sont composés d'unités mobiles communiquant via un média sans fil, sans la nécessité d'infrastructure physique. Dans ces réseaux, tous les nœuds coopèrent afin d'assurer la bonne gestion du réseau (contrôle, routage,…). Leurs spécificités fournissent des solutions pour construire des réseaux dynamiques, qui allient les avantages de la simplicité de construction à un coût relativement modeste. Ces réseaux amènent de nombreuses possibilités nouvelles, car ils ont une large gamme d'applications pratiques, qui comprennent et ne se limitent pas à:

- la gestion des situations d'urgence, telles que la détection et la surveillance des attaques chimiques, des attaques biologiques, des catastrophes naturelles, etc. ;
- la surveillance de la circulation automobile ;
- la détection des mouvements ennemis sur un champ de bataille ;
- les systèmes de surveillance et d'alarme ;
- etc.

et leur utilisation est de plus en plus répandue.

Pour toutes ces applications, le besoin de sécurité est évident, Ainsi, comme tout réseau de communication, le véritable potentiel des réseaux ad hoc ne peut être exploité sans envisager de résoudre de manière adéquate les problèmes de sécurité. Toutefois, en raison de la nature précaire des communications sans fil, des changements dynamiques dans la topologie du réseau, des contraintes en ressources des nœuds, ainsi que de leurs domaines d'application, ces réseaux posent encore de nos jours des défis majeurs en termes de sécurité. Assurer la sécurité des communications est l'un des enjeux majeurs dans de tels réseaux.

## 1. Contexte et Problématique de recherche

Dans les réseaux ad hoc, les techniques cryptographiques sont essentielles pour garantir les services de sécurité tels que la confidentialité, l'intégrité et l'authentification et ainsi assurer la sécurité des communications. L'utilisation des techniques cryptographiques pour sécuriser la communication entre deux entités, requiert que ces dernières aient une valeur secrète ou une clé. La nécessité d'avoir accès aux clés cryptographiques fait de la gestion des clés un service primordial pour la sécurité de n'importe quel système basé sur la communication.

La gestion des clés présente deux aspects importants : la distribution des clés, qui décrit comment diffuser des informations secrètes aux responsables, afin que des communications sécurisées puissent être initiées, et la révocation de clés, qui décrit comment supprimer les secrets pouvant avoir été compromis. La gestion des clés dans les réseaux ad hoc est confrontée à des contraintes majeures, qui se résument à l'absence d'infrastructure, aux contraintes en ressources des nœuds du réseau et surtout à la facilité de mise en œuvre des attaques. Dans ces réseaux, le risque de compromission doit généralement être considéré comme très élevé, due à la faible protection physique des nœuds et à leur facilité d'accès par des utilisateurs malveillants. Lorsqu'un nœud est compromis, les clés stockées dans celui-ci sont révélées à l'adversaire, ce qui affecte la communication sécurisée. Ces clés sont également souvent partagées par de nombreux nœuds du réseau. Par conséquent tant que les clés compromises ne sont pas révoquées de tous les nœuds du réseau, la communication sécurisée est affectée. Ainsi toute solution de gestion des clés dans les réseaux ad hoc doit disposer d'un mécanisme efficace de révocation de clés pour pouvoir adresser les menaces, qui pourront aboutir à la compromission de la confidentialité des clés, la compromission de l'authenticité des clés et à l'usage non autorisé de clés.

La problématique liée à la gestion des clés dans les réseaux ad hoc en général, dans les réseaux mobiles ad hoc (MANETs) et les réseaux de capteurs sans fil (WSNs) en particulier a déjà fait l'objet d'une multitude de publications [1-17]. Alors que (dans ce domaine, et dans ce type de réseaux,) la génération et la distribution de clés ont largement été étudiées, la révocation de clés a reçu relativement peu d'attention [17-21]. En effet, dans la plupart des mécanismes de gestion de clés proposés pour les MANETs et les WSNs, la question de la révocation de clés en cas de compromission de nœuds reste souvent non abordée, ou est insuffisamment traitée [22-27]. Un problème de sécurité vital comme celui-ci ne devrait-il pas bénéficier d'une reconnaissance appropriée et être considéré comme un domaine de recherche important, ou doit-il simplement être considéré comme un segment partiel de gestion des clés qui peut être négligé ?

De plus en plus de chercheurs se rendent compte de l'importance de cet aspect et tentent de répondre à cette question en proposant des solutions de révocation de clés. Toutefois malgré

l'utilisation de diverses techniques et approches par les concepteurs de protocoles de révocation de clés pour les MANETs et les WSNs, les problèmes non résolus restent multiples. La raison à cela n'est-elle pas l'absence d'une étude comparative exhaustive des protocoles et approches de révocation de clés pour les MANETs et les WSNs visant à faire ressortir les forces et faiblesses de chaque approche et de chaque protocole et les éventuelles possibilités d'exploiter ces résultats ?

L'étude et l'analyse menées dans cette thèse nous permettent d'obtenir la réponse à cette question puisqu'elles nous ont permis de mettre la lumière sur les défis existants encore non résolus par toutes ces propositions de solutions de révocation de clés et ont suscité les interrogations suivantes :

1. En ce qui concerne les MANETs, est-il possible de réduire la charge de communication et le délai associés à la propagation des informations pour la révocation des clés d'un nœud compromis ?
2. En ce qui concerne les WSNs est-il possible d'obtenir une solution de gestion de clés proposant un mécanisme de révocation, produisant un coût de communication plus réduit en termes de consommation énergétique et de délai, et offrant la possibilité d'un passage à l'échelle ?

Telles sont les trois questions de recherche nous nous proposons d'adresser dans ce travail.

## 2. Objectifs de recherche

L'objectif principal de cette thèse est de fournir de manière exhaustive une analyse des approches et solutions existantes pour une meilleure appréhension de la question liée à la révocation de clés dans les MANETs et les WSNs, qui conduira au développement des solutions adéquates en réponse aux questions posées.

Pour y parvenir, les sous-objectifs suivants sont alors visés :

- Montrer l'importance et la place qu'occupe la révocation de clés dans le processus de sécurisation des MANETs et des WSNs via la gestion des clés ;

- Montrer le déséquilibre existant dans le domaine de la recherche relative à la gestion de clés dans les MANETs et les WSNs, causé par (la négligence, le nombre peu élevé de travaux relatifs à) la question de la révocation de clés dans ces deux types de réseaux et les défis associés à la fourniture de solutions de révocation de clés pour ces derniers ;

- Etudier les différentes approches qui ont été adoptées pour la proposition de solutions de révocation de clés tant dans les MANETs que dans les WSNs ;

- ❖ Etudier et analyser les forces, les faiblesses de chaque approche en général et de chaque protocole en particulier ;

- ❖ Apporter des améliorations aux solutions de révocation de clés existantes pour les MANETs et les WSNs afin de les rendre plus sécurisées, plus performantes et donc mieux adaptées à leur contexte d'utilisation ;

- ❖ Donner de nouvelles orientations qui peuvent servir pour une meilleure prise en main du problème de la révocation de clés dans les MANETs et les WSNs.

## 3. Méthodologie de travail

Pour atteindre ces objectifs, dans la première partie de notre travail :

- ✓ Nous décrivons les caractéristiques générales des MANETs et des WSNs en mettant l'accent sur les propriétés qui font de l'application de la sécurité une tâche complexe ;

- ✓ Nous présentons par la suite les principaux aspects couverts par la sécurité dans les MANETs et WSNs, à savoir les potentielles attaques de sécurité, les services et les mécanismes de sécurité en insistant sur ceux qui font appel à la cryptographie. Ensuite, nous esquissons le processus de gestion des clés en général, et dans les MANETs et les WSNs en particulier, en donnant un aperçu sur les différentes étapes qu'il peut inclure et en montrant la place importante qu'occupe la révocation de clés dans ce processus ;

- ✓ Nous présentons enfin dans cette première partie une revue de la littérature, qui porte sur la révocation de clés dans les MANETs et les WSNs en donnant pour chaque type de réseau une classification des protocoles existants, et en analysant le fonctionnement de chaque protocole pour faire ressortir les différentes approches qui ont été adoptées.

Toujours dans le but d'atteindre nos objectifs dans la deuxième partie de notre travail nous présentons nos différentes contributions :

- ✓ La première consiste en une étude approfondie se traduisant par une analyse de sécurité et performance et une comparaison des différentes propositions de solutions, ceci afin de faire ressortir les forces et les faiblesses de chaque catégorie de solution et de chaque protocole, mais surtout de mettre en évidence les problèmes qui n'ont pas été résolus ;

- ✓ La deuxième contribution se focalise sur les insuffisances relatives aux propositions déjà faites pour les MANETs. Elle consiste en la fourniture d'un mécanisme efficace de révocation de certificat. Notre proposition modifie le mécanisme de révocation de clés proposé dans [28] pour les systèmes basés sur l'identité afin de l'adapter aux protocoles basés sur PKI ;

- ✓ La troisième et dernière contribution, destinée aux WSNs, propose d'optimiser le protocole de gestion de clés CL-EKM proposé par [29] en réduisant la consommation énergétique et le délai relatifs aux échanges entre la station de base et les nœuds CH chef de cluster. Ceci afin de mieux répondre aux contraintes des WSNs et ainsi adresser les limites relatives au protocole de révocation associé ;

- ✓ Nous montrons analytiquement la validité et l'efficacité de nos propositions par une étude évaluative et comparative de nos propositions par rapport aux deux solutions respectives qui ont servi de référence, et montrons des gains clairs aussi bien en termes de sécurité qu'en termes de performance.

## 4. Résultats Escomptés

A l'issue nos travaux nous comptons obtenir les résultats suivants :

- ✓ Un comparatif des approches et protocoles de révocation de clés proposées pour les MANETs faisant ressortir leurs avantages et inconvénients ;

- ✓ Un comparatif des approches et protocoles de révocation de clés proposées pour les WSNs faisant ressortir leurs avantages et inconvénients ;

- ✓ Une solution de révocation de certificat pour les MANETs, résistante aux potentielles attaques et produisant une charge de communication et un délai liés à la propagation des informations de révocation plus réduits ;

- ✓ Un protocole de gestion de clé pour les WSNs proposant un mécanisme de révocation de clés, résistant aux potentielles attaques, produisant une consommation énergétique et un délai de transfert des échanges plus réduits et une possibilité de passage à l'échelle ;

- ✓ Des pistes pour trouver des moyens efficaces de gestion du problème que pose la révocation de clés dans les MANETs et les WSNs et combler le déséquilibre existant dans le domaine de la gestion des clés dans les MANETs et des WSNs.

## 5. Organisation du manuscrit

Ce manuscrit, constitué d'une introduction générale, de sept chapitres et d'une conclusion générale, est organisé en deux parties. La première partie qui comporte les trois premiers chapitres présente une revue de la littérature, et la seconde partie comportant les chapitres restants décrit nos différentes contributions. Nous détaillons ci-dessous le contenu des chapitres.

Dans le premier chapitre, nous présentons les concepts généraux des MANETs et des WSNs en exposant leurs spécificités et en mettant l'accent sur les aspects qui rendent l'application de la sécurité une tâche difficile dans ces deux types de réseaux.

Dans le deuxième chapitre, nous donnons d'abord un aperçu sur la sécurité dans les MANETs et les WSNs. Nous introduisons ensuite la question liée à la gestion des clés dans ces deux types de réseaux avant de montrer le rôle et l'importance de la révocation de clés dans le processus de sécurisation de ces réseaux.

Dans le troisième chapitre, nous donnons une classification des différents protocoles de révocation de clés utilisés dans les MANETs et les WSNs et décrivons leur fonctionnement.

Dans le quatrième chapitre, nous présentons notre première contribution en effectuant une analyse des différentes approches et solutions de révocation de clés proposées pour les MANETs en nous basant sur les critères de sécurité, de performance et d'applicabilité.

Dans le cinquième chapitre, nous présentons notre deuxième contribution fournissant une analyse des différentes approches et solutions de révocation de clés proposées pour les WSNs en nous basant toujours sur les critères de sécurité, de performance et d'applicabilité.

Le sixième chapitre décrit et évalue le protocole de révocation de certificat *SECRET*, que nous proposons dans le but d'apporter des solutions qui adressent certaines insuffisances soulignées dans le chapitre 4, suite à l'analyse des protocoles de révocation proposés pour les MANETs.

Le dernier chapitre décrit et évalue la solution *ECL-EKM* que nous proposons afin d'adresser les insuffisances soulignées dans le chapitre 5 relatives au mécanisme de révocation de clés fourni pour WSNs par le protocole de gestion de clés *CL-EKM.*

# Première Partie :

# REVUE DE LA LITTERATURE

# Chapitre 1 CONCEPTS GENERAUX DES MANETs ET WSNs

## 1.1 Introduction aux réseaux sans fil

Ces dernières années, les réseaux utilisés traditionnellement se sont montrés inadaptés à résoudre les problèmes posés par notre nouveau style de vie collective. Si des utilisateurs doivent être connectés à un réseau par des câbles physiques, leurs déplacements en sont dramatiquement réduits. Une connexion sans fil n'impose pas autant de restrictions et accorde une plus grande liberté de mouvement à l'utilisateur du réseau, c'est-à-dire une grande mobilité. Ainsi, les technologies sans fil commencent à gagner du terrain sur les réseaux traditionnels câblés [30-32].

La mobilité engendre pour les terminaux des problèmes spécifiques à cet environnement :

- une perte fréquente de connexion ;
- un faible débit de communication ;
- des ressources systèmes modestes ;
- une capacité d'énergie limitée pour les mobiles.

Dans la section suivante, nous présentons une brève introduction sur les types de réseaux sans fil ainsi, que les différentes technologies qu'ils implémentent.

### 1.1.1 Les types de réseaux sans fil

Les réseaux sans fil sont en plein développement du fait de la flexibilité de leur interface, qui permet à un utilisateur de changer de place tout en restant connecté. Contrairement à l'environnement statique, l'environnement mobile permet aux unités de calcul une libre mobilité.

Il existe principalement deux catégories de types de réseaux sans fil que sont :

- Les réseaux sans fil avec infrastructure fixe (SBS ou *Basic Service Set*) ;
- Les réseaux mobiles ad hoc (ISBS ou Independent *Basic Service Set*).

#### 1.1.1.1 Les réseaux sans fil avec infrastructure fixe (SBS ou Basic Service Set)

Dans ce type de réseau les stations de base (SBs) reliées entre elles par un réseau filaire assurent la couverture d'une zone, et prennent en charge les mobiles dans leur voisinage. Les SBs sont fixes et munis d'une interface sans fil pour la communication directe avec les terminaux mobiles localisés dans une zone géographique limitée, appelée cellule. La cellule est contrôlée par un point d'accès (AP, *Access Point*) qui coordonne les transmissions et sert de pont entre le réseau câblé et le WLAN

(*Wireless Local Area Network*). Un inconvénient des réseaux cellulaires est que si le mobile n'a pas de station de base à sa portée, il ne peut plus se connecter [30-32].

La figure 1.1 illustre la structure d'un réseau sans fil avec infrastructure.

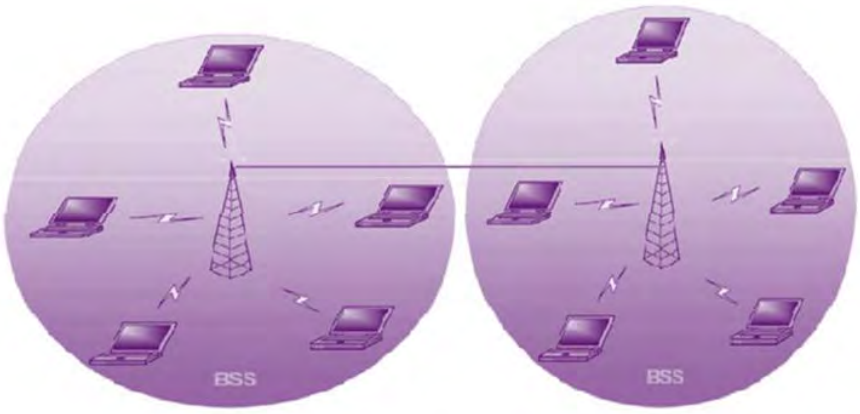



**Figure 1. : Structure d'un réseau sans fil en mode avec infrastructure**

#### 1.1.1.2 Les réseaux mobiles ad hoc (ISBS ou Independent Basic Service Set)

Dans un réseau mobile ad hoc, constitué de terminaux munis d'une interface de communication radio, un nœud du réseau peut communiquer directement avec ses voisins, c'est-à-dire ceux qui sont à portée de communication de sa propre interface, et peut faire office de routeur pour les nœuds mobiles du réseau. Ce type de réseaux, appelé aussi MANET pour Mobile Ad hoc NETwork, est utile quand aucune connexion filaire n'est disponible, par exemple lors d'interventions sur le site d'une catastrophe, lors d'une opération militaire, et plus généralement quand le déploiement rapide d'un réseau est nécessaire [30-32]. Dans ces réseaux, nous n'avons plus de station de base, car ce sont les nœuds intermédiaires qui servent de passerelles ou de relais pour les autres nœuds mobiles du réseau.

Un réseau ad hoc peut être défini comme étant une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide d'une quelconque administration centralisée ou de tout support fixe. Pour pouvoir fonctionner sur un réseau étendu, ce mode doit être associé à un protocole de routage permettant à un terminal de communiquer avec un autre terminal éloigné par l'intermédiaire de terminaux faisant office de routeur. Ce mode de fonctionnement accentue considérablement un des problèmes majeurs dans les réseaux ad hoc, qui consiste à trouver les routes optimales et fiables entre les nœuds mobiles.

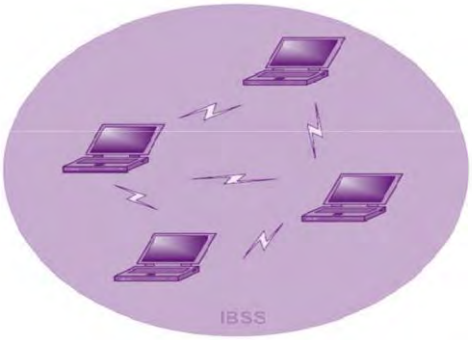



**Figure 1. : Structure d'un réseau sans fil en mode ad hoc**

Le mode ad hoc multi-sauts présente des caractéristiques qui peuvent représenter des avantages, mais également des inconvénients. En effet, comparés avec le mode de communication avec stations de base, un mobile émet plus de messages en mode ad hoc qu'en mode infrastructure, puisqu'il doit à la fois transmettre ses propres informations, mais également les données des autres mobiles pour lesquels il fait office de routeur. La diminution de portée de communication permet d'économiser beaucoup d'énergie [30-32]. Nous pourrions donc penser que ce mode est plus gourmand en énergie. Néanmoins, les portées de communication peuvent être largement réduites en mode ad hoc (à titre d'exemple, on parle de dix mètres en mode ad hoc contre cent mètres en mode infrastructure). De plus, diminuer les puissances d'émission permet également de limiter les risques de collisions entre les communications, ce qui diminue le nombre de retransmissions dues aux pertes de paquets, et donc de faire plus d'économies d'énergie.

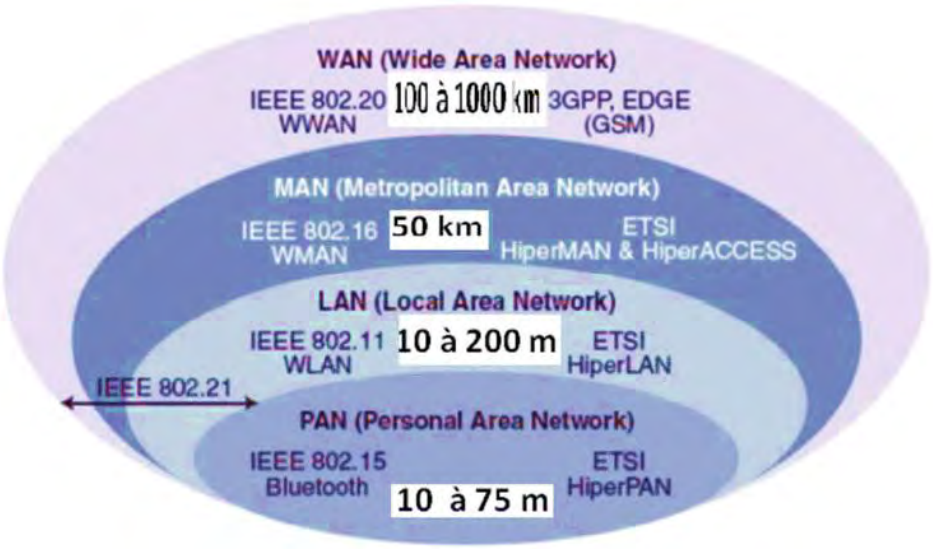
### 1.1.2 Les gammes de technologies sans fil

Depuis la fin du 20e siècle, le monde a de plus en plus besoin de mobilité, de l'accès et du partage de l'information. Ce besoin de mobilité se concrétise par la miniaturisation des périphériques qui sont caractérisés par une source d'énergie limitée (PDA, appareil photo numérique, téléphone portable, ...). Cependant, au début de leurs créations, ces différents appareils ne pouvaient communiquer entre eux ou se connecter à des réseaux informatiques. Pour assurer l'interopérabilité ces appareils implémentent les technologies des réseaux sans-fil.

Plusieurs gammes de technologies sont actuellement commercialisées. Les principales normes sont:

- IEEE 802.15, pour les petits réseaux sans fil personnels aussi appelés PAN (*Personal Area Network*) d'une dizaine de mètres de portée ;
- IEEE 802.11, ou Wi-Fi, pour les réseaux WLAN (*Wireless Local Area Network*) ;
- IEEE 802.16, pour les réseaux WMAN (*Wireless Metropolitan Area Network*) atteignant plus de dix kilomètres ;
- IEEE 802.20, pour les réseaux WWAN (*Wireless Wide Area Network*), c'est-à-dire les grands réseaux [30 -32].

La figure 1.3 illustre les différentes catégories de réseaux suivant leur étendue [33].




:

**Figure 1. : Différentes catégories de réseaux sans fil [33]**

#### 1.1.2.1 Les Wireless Personal Area Network (IEEE 802.15)

La norme IEEE 802.15 sert à créer de minuscules réseaux sans fil, appelés WPAN pour Wireless Personnel Area Network. Ces réseaux ont des portées qui peuvent s'étendre jusqu'à des dizaines de mètres et sont prévus pour interconnecter différents périphériques autonomes entre eux. Cette norme est souvent appelée Bluetooth, mais ce n'est qu'une application de cette norme. En effet, la norme IEEE 802.15.1 a été adaptée à partir des spécifications Bluetooth déjà existantes. Mais la norme IEEE 802.15 est divisée en quatre parties, qui normalisent des gammes de produits en parallèle [30-32] :

- IEEE 802.15.1, le plus connu, prend en charge la norme Bluetooth, aujourd'hui largement commercialisée, elle exploite la bande de fréquence 2,4 GHz. Elle permet d'obtenir un débit de 1 Mbit/s ;
- IEEE 802.15.2 propose des recommandations pour l'utilisation de la bande de fréquence 2.4 GHz (fréquence utilisée également par le Wifi). Avec un débit d'environ 3 à 4 Mbit/s, elle peut atteindre une portée de 15 à 20 mètres ;
- IEEE 802.15.3 est un standard visant à proposer du haut débit (20 Mbit/s) avec la technologie Bluetooth. Elle définit la norme UWB (*Ultra-Wide Band*), qui met en œuvre une technologie très spéciale, caractérisée par l'émission à une puissance extrêmement faible, dans le bruit ambiant, mais sur pratiquement l'ensemble du spectre radio (entre 3,1 et 10,6 GHz). Elle peut atteindre une distance de 10 mètres ;

- IEEE 802.15.4 utilise la bande du spectre sans licence d'utilisation des fréquences de 2,4 GHz en visant un faible coût de mise en place et d'utilisation. Elle peut atteindre une distance de 10 à 75 mètres, avec un débit d'environ 250 kbit/s.

#### 1.1.2.2 Les Wireless Local Area Network (IEEE 802.11)

La norme IEEE 802.11 sert à créer des réseaux sans fil, avec une portée pouvant aller jusqu'à des centaines de mètres. Il existe de nombreuses normes dérivées de celle-ci. Les trois plus connues sont la norme IEEE 802.11b, qui est à l'origine des réseaux à 11 Mbit/s dans la bande de fréquence des 2,4 GHz, sa portée peut atteindre 50 à 200 mètres. La norme IEEE 802.11a qui offre un débit de 54 Mbit/s dans la bande de fréquence des 5,3 GHz avec une portée de 10 mètres. La norme IEEE 802.11g qui est un compromis avec les deux normes précédentes en offrant un débit de 54 Mbit/s dans la bande de fréquence des 2,4 GHz avec une portée pouvant atteindre 25 à 75 mètres. Il est intéressant de remarquer que certaines normes dérivées apportent de nombreuses améliorations. Ainsi, la norme IEEE 802.11i met l'accent sur la sécurité des données plus particulièrement l'authentification et la confidentialité. La norme IEEE 802.11n doit pouvoir atteindre des débits de 100 Mbit/s sur la bande de fréquence des 2,4 GHz. La norme IEEE 802.11e veut améliorer la qualité de service, surtout dans la transmission de la voix et de la vidéo [30-32].

#### 1.1.2.3 Les Wireless Metropolitan Area Network (IEEE 802.16)

Destiné aux réseaux sans fil métropolitains WMAN (*Wireless Metropolitan Networks*), 802.16 est développé par le groupe industriel WiMAX (*World wide Interoperability for Microwave Access*). Il existe plusieurs versions de cette norme. La norme WIMAX 802. 16 fonctionnant dans la bande des fréquences 10 à 66 GHz a besoin d'antennes en vue directe.

La norme IEEE 802.16a fonctionne quant à elle dans la bande des fréquences de 2 à 11 GHz mais ne requiert pas de vue directe pour les antennes. Cette dernière devrait permettre une desserte sur un rayon de 50 kilomètres et atteindre une bande passante totale de 70 Mbit/s. La norme IEEE 802.16c définit les profils d'utilisation dans la bande des fréquences 10 à 66 GHz comme le transport de flux audio/vidéo, la téléphonie numérique, ou les trames ATM. La norme IEEE 802.16e ajoute la mobilité à ces réseaux (*Handover*). Enfin, la norme IEEE 802.16.2 permet l'inter-compatibilité entre toutes les normes 802.16.

Toutes les normes IEEE 802.16 incluent de base la notion de Qualité de Service permettant par exemple le transport de la voix ou de la télévision [30-32].

**Tableau 1. : Tableau récapitulatif des normes sans fil**

| Norme | Fréquence | Débit | Portée |
|---|---|---|---|
| 802.11a | 5 GHz. | 54 Mbit/s en théorie et 30 Mbit/s réels | 10 m |
| 802.11b | 2.4 GHz | 11 Mbit/s, 6.5 Mbit/s | ~50 m à 200m |
| 802.11g | 2.4 GHz | 54 Mbit/s en théorie et 30 Mbit/s réels | ~25 m à 75m |
| IEEE 802.15.1 | 2,4 GHz | 1 Mbit/sec | 10m |
| IEEE 802.15.2 | 2,4 GHz | 3 à 4 Mbit/s | 15 à 20 m |
| IEEE 802.15.3 | 3,1 et 10,6 GHz | 20 Mbit/s | 10 m |
| IEEE 802.15.4 | 2,4 GHz | 250 kbit/s | 10 à 75 m |
| IEEE 802.16a | 2 à 11 GHz | 70 Mbit/s | 50 km |
| IEEE 802.16c | 10 à 66 GHz | 70 Mbit/s | 50 km |
| IEEE 802.16e | 5 à 6 GHz | 70 Mbit/s | 50 km |
| IEEE 802.20 | 3,5 à 5 GHz | 1 à 4 Mbit/s | 100 à 1000 km |
| IEEE 802.22 | 54 à 862 GHz | Non défini | 40 à 100 km |

## 1.2 Les Réseaux Mobiles Ad Hoc (MANETs)

Dans cette section nous présentons une description générale des réseaux ad hoc ainsi que les défis associés à leur sécurisation. Nous limitons notre attention aux propriétés des MANETs qui rendent l'application de la sécurité une tâche difficile.

### 1.2.1 Définition d'un réseau mobile Ad hoc

Un réseau mobile ad hoc (Mobile Ad hoc Network (MANET)) est un réseau maillé temporaire constitué par une collection de nœuds sans fil et mobiles, sans l'aide d'une infrastructure préétablie, utilisée pour exécuter les fonctions de base de gestion de réseau comme l'acheminement des paquets. La topologie d'un réseau ad hoc est représentée par la figure 1.4.

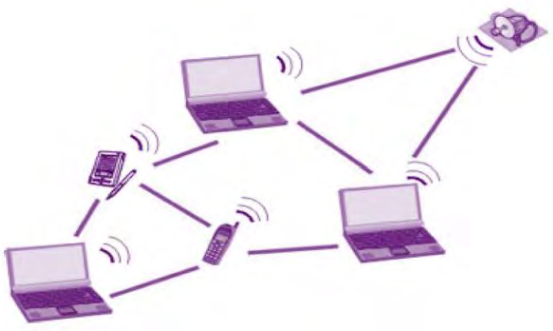


**Figure 1. : Topologie d'un réseau mobile ad hoc**

Cette topologie montre qu'un réseau Ad Hoc est constitué d'un regroupement d'une grande population d'unités de calcul portables (*laptops*, *PDA*, *Smartphones*,…) se déplaçant dans un territoire quelconque, avec leur interface sans fil pour seul moyen de communication. La miniaturisation des périphériques peut engendrer des problèmes d'adaptation des protocoles de routage.

### 1.2.2 Caractéristiques des réseaux mobiles Ad hoc

Les réseaux mobiles Ad-hoc sont caractérisés par :

- **L'absence d'infrastructure :** qui fait que les nœuds sont eux-mêmes responsables de l'établissement et du maintien de la connectivité du réseau. En l'absence de toute entité fixe, il devient difficile de mettre en place une infrastructure à clé publique avec la définition d'une autorité de certification centralisée. Ceci pose un problème d'authentification, ce qui peut engendrer des attaques par usurpation ou mascarade dans le réseau [34-40] ;
- **Une topologie dynamique due à la mobilité des nœuds :** les nœuds du réseau sont libres de se déplacer indépendamment les uns des autres. Ainsi, la topologie tend à changer rapidement et de manière imprévisible. En outre, ce changement peut conduire à des débordements dans les tables de routage créant ainsi des dénis de service [34-40] ;
- **Une bande passante limitée :** qui influe considérablement sur le volume des informations échangées. Pour l'instant les liaisons sans fil offrent significativement moins de capacité de débits que les liaisons câblées. D'autre part, le rendement obtenu sur des communications sans fil – si l'on considère les effets de l'accès au média et les phénomènes d'atténuation du signal, de bruit ou d'interférence, etc. – est sensiblement inférieur au débit théorique maximal que permettent les antennes radio. Une autre conséquence de la faible capacité des liens est que la congestion est très courante dans ces réseaux. En effet, les besoins des applications actuelles excédent fréquemment les capacités du réseau. Aussi, les attaques par déni de service ont-elles plus d'impact dans les réseaux ad hoc, puisque la bande passante est aisément saturée [34-39] ;
- **Des contraintes en énergie :** les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries. Les nœuds de ces réseaux sont la plupart du temps des terminaux mobiles et s'appuient sur des batteries dont la capacité est limitée par rapport aux calculs effectués. Un besoin important pour ces nœuds est donc l'économie d'énergie. D'autre part, les performances de ces nœuds, qu'il s'agisse de la puissance des processeurs ou des capacités de stockage, sont plus restreintes que dans les réseaux fixes. Ces contraintes font que les attaques par déni de service (épuisement des batteries, saturation de la mémoire, etc.) sont facilement menées dans ce type de réseau [34-40] ;
- **L'auto configuration :** cette propriété permet aux nœuds de s'intégrer facilement dans un réseau. Elle facilite la gestion du réseau car l'interconnexion des éléments ne nécessite qu'un

minimum d'intervention technique externe. Cette fonctionnalité est de plus en plus nécessaire pour un déploiement à grande échelle des réseaux sans fil ad hoc. L'intégration facile d'un nœud dans le réseau peut engendrer des problèmes d'identification, ce qui aboutirait à des attaques par usurpation d'identité (d'adresse) ou mascarade dans ce type de réseau [34-39] ;

- **L'hétérogénéité des nœuds :** un nœud mobile peut être équipé d'une ou plusieurs interfaces radio ayant des capacités de transmission variées et opérant dans des plages de fréquences différentes. De plus, les nœuds peuvent avoir des différences en termes de capacité de traitement (CPU, mémoire), de logiciel et de mobilité. Cette hétérogénéité des nœuds engendre des problèmes d'interopérabilité surtout au niveau du canal. Ceci peut être utilisé par les attaquants pour provoquer le brouillage du canal [34-39] ;
- **Le besoin en coopération :** en l'absence de routeur, chaque participant peut être amené à acheminer des paquets aux autres nœuds du réseau. Cependant, l'un de ces participants peut décider de ne pas relayer les paquets, soit dans le but « égoïste » d'économiser ses propres ressources, soit dans un but volontairement malveillant. Les nœuds intermédiaires qui refusent de jouer pleinement leur rôle de relais peuvent alors entrainer un dysfonctionnement du réseau. Par conséquent, un nœud attaquant peut créer des attaques par déni de service dans ce type de réseau en décidant de ne pas participer à l'élaboration des routes ou de ne pas relayer les paquets [34-39] ;
- **Une sécurité physique limitée :** liée à la vulnérabilité du support de communication classique en raison des attaques par écoute, par usurpation d'identité et par déni de service. Cela se justifie par les contraintes et limitations physiques qui sont dues à la propagation des ondes dans l'espace, ce qui fait que la confidentialité et la disponibilité des données à transférer doivent être assurées [34-36].

### 1.2.3 Domaines d'applications des MANETs

D'une façon générale, les réseaux ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure filaire est trop contraignant, soit parce que difficile à mettre en place, soit parce que la durée d'installation du réseau ne justifie pas le recours au câblage à demeure.

Les premières applications des MANETs concernaient les communications et les opérations dans le domaine militaire. Cependant, avec l'avancement des recherches dans le domaine des réseaux, et l'émergence des technologies sans fil (ex : *Bluetooth*, IEEE 802.11 et *Hiperlan*), d'autres applications civiles sont apparues. Ainsi on distingue [41] :

- ***Les services d'urgence :*** opérations de recherche et de secours des personnes, tremblements de terre, incendies, inondations, dans le but de remplacer l'infrastructure filaire ;

- ***Le travail collaboratif et les communications dans des entreprises ou bâtiments :*** dans le cadre d'une réunion ou d'une conférence par exemple ;
- ***Home network :*** partage d'applications et de communications des équipements mobiles.
- ***Applications commerciales :*** pour un paiement électronique distant (taxi) ou pour l'accès mobile à l'Internet, ou service de guide en fonction de la position de l'utilisateur ;
- ***Réseaux de capteurs :*** pour des applications environnementales (climat, activité de la terre, suivi des mouvements des animaux, etc.) ou domestiques (contrôle des équipements à distance) ;
- ***Réseaux en mouvement :*** informatique embarquée et véhicules communicants.

### 1.2.4 Les Problèmes et les contraintes des MANETs

Vu les rôles multiples que chaque nœud participant joue dans un MANET, les grandes sources de problèmes des réseaux mobiles ad hoc sont la grande mobilité des nœuds, la limite d'énergie et la faible sécurité.

- **La mobilité** [42] **:** La mobilité des nœuds conduit à des changements fréquents de la topologie des réseaux. Cette dernière peut être la cause de la déconnexion fréquente, et ce phénomène est d'autant plus important lors des communications de groupe au sein des réseaux sans fil ad hoc. La mobilité des nœuds représente un facteur clé au niveau de la qualité de services (transmission des paquets), de la bonne exploitation de la bande passante du réseau, ainsi que de la sécurisation qui reste un grand défi pour ce genre de réseaux ;
- **L'énergie** [43] **:** Vu le manque d'infrastructure, le manque de points de raccordement entre les membres ainsi que la topologie dynamique, certains nœuds participants jouent le rôle de routeurs pour assurer le routage du grand flux de paquets circulants sur le réseau et l'acheminement des clés de cryptage. Dans le cas où la position de ces nœuds est stratégique ceci peut causer une grande consommation d'énergie et donc l'épuisement rapide de leur batterie. Avec la liberté des nœuds à se déplacer dans le réseau, l'échange des paquets de contrôle grandit au fur et à mesure. Plus la topologie change vite et profondément, plus la charge de contrôle du trafic sera importante. Ces nœuds peuvent rapidement épuiser leur ressource énergétique ;
- **La faible sécurité :** La sécurisation du trafic de contrôle, la sécurisation des informations et la gestion du routage au sein des réseaux ad hoc représentent un grand défi qui est causé par le manque de l'infrastructure fixe [43]. Dans le cas des réseaux ad hoc, les problèmes de sécurisation sont tellement complexes que l'ajout de nouveaux nœuds mobiles au réseau nécessite un contrôle strict afin de détecter et d'arrêter les nœuds "intrus" qui détourneraient ou perturberaient le fonctionnement même du routage.

## 1.3 Les réseaux de capteurs sans fil (WSNs)

Un capteur (senseur) est un système qui sert à détecter un phénomène physique afin de le représenter sous forme de signal souvent électrique. Les capteurs sont des petits appareils dotés d'une batterie, capables de communiquer entre eux et de détecter des événements s'ils se trouvent à l'intérieur de leur rayon de perception. Ils sont dotés de mécanismes leur permettant de relever des informations sur leur environnement. La nature de ces informations varie très largement selon l'utilisation qui est faite du capteur : ce dernier peut tout aussi bien faire des relevés de température, d'humidité ou d'intensité lumineuse. Un capteur possède également le matériel nécessaire pour effectuer des communications sans-fil par ondes radio [44].

### 1.3.1 Définition d'un réseau de capteurs sans-fil

Les réseaux de capteurs sans-fil (*Wireless Sensor Network (WSN)*) sont considérés comme un type spécial de MANET qui inclut des nœuds capteurs avec des capacités de calcul et de communication limitées déployés en grand nombre, en particulier dans les zones hostiles [45]. Les nœuds capteurs sont capables d'accomplir trois tâches complémentaires : le relevé d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs. L'ensemble de ces capteurs, déployés pour une application, forme un réseau de capteurs.

Le but d'un réseau de capteurs est de surveiller une zone géographique, et parfois d'agir sur celle-ci (il s'agit alors de réseaux de capteurs-actionneurs). On peut citer comme exemples un réseau détecteur de feu de forêt, ou un réseau de surveillance de solidité d'un pont après un tremblement de terre. Le réseau peut comporter un grand nombre de nœuds (des milliers). Les capteurs peuvent être placés de manière plus ou moins aléatoire (par exemple par largage depuis un hélicoptère) dans des environnements pouvant être dangereux. Toute intervention humaine après le déploiement des nœuds capteurs est la plupart du temps exclue, le réseau doit donc s'autogérer. Afin que les nœuds capteurs travaillent d'une façon coopérative, les informations recueillies sont partagées entre eux par communication radio (ondes électromagnétiques). Le choix du lien radio plutôt que du lien filaire permet un déploiement facile et rapide dans un environnement pouvant être inaccessible pour l'être humain [44].

### 1.3.2 Objectifs de base des WSNs

Les objectifs de base des réseaux de capteurs sans-fil dépendent généralement des applications, cependant les tâches suivantes sont communes à plusieurs applications :

- Déterminer les valeurs de quelques paramètres suivant une situation donnée. Par exemple, dans un réseau environnemental, on peut chercher à connaître la température, la pression

atmosphérique, la quantité de la lumière du soleil, et l'humidité relative dans un nombre de sites, etc ;

- Détecter l'occurrence des événements spécifiques et estimer les paramètres des événements détectés. Dans les réseaux de contrôle de trafic, on peut vouloir détecter le mouvement de véhicules à travers une intersection et estimer la vitesse et la direction du véhicule ;
- Classifier l'objet détecté : dans un réseau de trafic, un véhicule est-il une voiture, un bus, etc. ?

### 1.3.3 Types de WSNs

Les WSNs peuvent être classifiés selon deux catégories :

1. Le modèle dynamique de réseau :

- Le réseau est constitué d'un ensemble de capteurs mobiles évoluant dans un environnement statique. Le but de tels réseaux est la plupart du temps l'exploration de zones inaccessibles ou dangereuses. Les travaux de recherche sont souvent orientés robotique, les nœuds jouant à la fois le rôle de capteur et d'actionneur ;
- Ou bien le réseau est constitué de capteurs fixes servant à la surveillance d'occurrence d'évènements sur une zone géographique [46]. Ici, le réseau n'effectue que la surveillance, les données mesurées sont transmises en mode multi-sauts à un nœud spécifique appelé « puits» qui est chargé, après réception, de mettre en œuvre les actions nécessaires. Ce puits peut être connecté, de manière filaire par exemple, à un autre réseau.

2. Le modèle de délivrance de données :

- Les capteurs transmettent périodiquement les informations recueillies (délivrance de données continue) ;
- Le capteur transmet des informations à la détection d'un évènement (délivrance de données basée évènement (*event-driven*)) ;
- C'est à l'utilisateur de lancer une requête pour avoir l'information (en anglais "*observer-initiated*") ;
- Délivrance hybride de données, où on trouve les différentes délivrances citées auparavant en même temps.

### 1.3.4 Architecture de base d'un capteur

Un capteur est composé de quatre éléments principaux :

- Un élément qui se charge de mesurer l'environnement extérieur (unité de capture) ;
- Une unité de calcul ;

- Un élément émetteur/récepteur ;
- Une alimentation.

Trois composants additionnels peuvent être implantés dans un capteur :

- Un système de recherche d'emplacement ;
- Un générateur d'alimentation ;
- Une unité mobile (permettant de faire bouger le capteur).

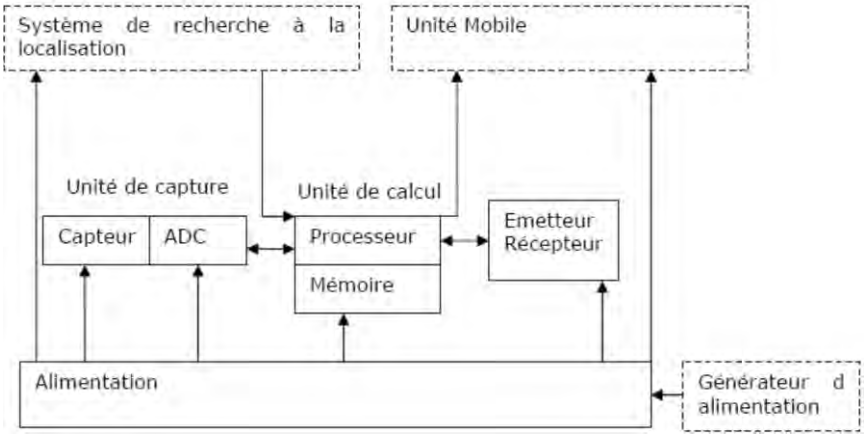



**Figure 1. : Architecture de base d'un capteur [44].**

L'unité de capture (où élément capteur) est composée de deux sous éléments :

- Le capteur récupérant des données analogiques ;
- Un convertisseur faisant passer les données analogiques du capteur à des données numériques (appelée ADC pour *analog to digital convertor*) envoyées à une unité de calcul.

L'unité de calcul regroupant :

- Un processeur ;
- Une unité de mémoire réduite.

Il permet de stocker les données, exécute les tâches de perception qui lui sont assignées.

L'Emetteur/Récepteur est l'élément permettant de connecter le capteur au réseau.

L'alimentation est la source d'énergie pour le capteur. Comme tout dispositif embarqué, le capteur dispose d'une alimentation autonome telle qu'une batterie.

### 1.3.5 Structure d'un réseau de capteur sans fil

La structure du WSN est montrée dans la figure 1.6. L'utilisateur accède à distance aux données capturées à travers un nœud appelé le nœud gestionnaire de tâche "*Task Manager Node*". Le nœud

directeur de tâche est relié à l'Internet ou au satellite à travers un nœud destinataire appelé "puits" (*sink* en anglais). Ce dernier agit en tant que passerelle pour le WSN, c'est à dire qu'il relie des réseaux de capteurs à d'autres réseaux. Ce nœud est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises aux capteurs via des messages de requêtes. Il a également d'autres capacités de traitement de l'information pour une transformation ultérieure, s'il y a lieu. Les nœuds capteurs sont habituellement dispersés dans une zone de capture appelée champ de captage (ou champ d'intérêt).

Les nœuds capteurs rassemblent les données et les transmettent au destinataire. De cette manière, les utilisateurs peuvent rechercher l'information dans les nœuds destinataires pour surveiller et commander l'environnement à distance. Notons qu'un WSN peut contenir plusieurs nœuds puits diffusant des intérêts différents (ce sont la description des données requises par le nœud destinataire en utilisant une appellation combinée attribut-valeur). Par exemple, un nœud puits peut demander à tous les capteurs se trouvant dans la région nord du champ de captage d'envoyer un rapport de température à chaque minute, pendant qu'un autre peut être intéressé seulement par les hautes températures dans la région sud. Par conséquent, un capteur doit pouvoir stocker toutes les requêtes reçues, et les traiter séparément [47].

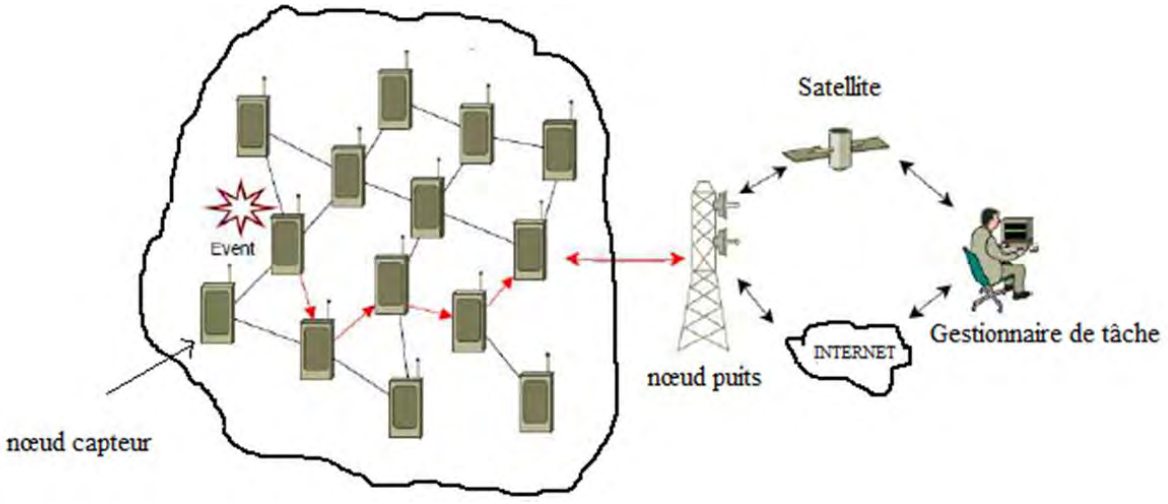



**Figure 1. : Structure d'un réseau de capteurs sans-fil.**

### 1.3.6 Facteurs et Contraintes Conceptuelles du Réseau et des Protocoles

La conception des WSNs, leurs protocoles et algorithmes sont guidés par plusieurs problèmes tels que :

- **L'environnement :** les capteurs sont souvent déployés en masse dans des endroits hostiles tels que des champs de bataille au-delà des lignes ennemies, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés, etc. Par

conséquent, ils doivent pouvoir fonctionner de manière autonome (sans surveillance) dans des régions géographiquement éloignées ou inaccessibles ;

- **Le modèle de livraison des données : S**elon l'application, le modèle de livraison des données au nœud PUITS peut être continu, périodique, commandé par événement, commandé par requête ou hybride ;
- **Le passage à l'échelle :** Le nombre de nœuds déployés pour un projet peut atteindre le million. Un nombre aussi important de nœuds engendre un trafic énorme dans le réseau, ce qui entraine des congestions et des erreurs de communication. Un tel déploiement, nécessite que le protocole utilisé pour la communication soit capable de détecter les erreurs et de contrôler le flux, et nécessite aussi que le nœud PUITS soit équipé d'une capacité de stockage suffisante pour accueillir les informations reçues à partir des nœuds capteurs ;
- **La topologie du réseau :** Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance se fait en trois phases :
    - Déploiement ;
    - Post-déploiement (les capteurs peuvent bouger, ne plus fonctionner, etc.) ;
    - Redéploiement des nœuds additionnels.
- **La Connectivité :** La densité élevée des nœuds dans les réseaux de capteurs les empêche d'être complètement isolés les uns des autres. Ceci, cependant, n'empêche pas la topologie du réseau d'être variable. En outre, la connectivité dépend de la distribution aléatoire des nœuds ;
- **La tolérance aux fautes :** Certains nœuds capteurs peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, d'une défaillance matérielle ou d'une interférence. La tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau sans interruption en cas de défaillance d'un de ses capteurs ;
- **La tolérance aux intrusions :** L'absence d'une protection physique des nœuds capteurs ainsi que la nature des liens sans-fil utilisés pour la communication, rend le réseau vulnérable aux attaques malveillantes. La tolérance aux intrusions implique la tolérance aux vulnérabilités, dont certaines sont inévitables pour améliorer à la fois la sécurité et l'utilisation du réseau ;
- **Le média de transmission :** Afin d'être installés sans difficulté dans des zones ciblées et sans induire d'importants coûts de câblage, les capteurs utilisent des liens radiofréquence pour communiquer entre eux au sein du réseau ;
- **Les contraintes matérielles :** En plus de leurs petites dimensions, les capteurs subissent de fortes contraintes, notamment d'énergie, de calcul et de stockage :
    - *Dimension* : la taille réduite des capteurs peut présenter de nombreux avantages en fonction de l'utilisation prévue du système, et elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée par la petite taille de ces derniers ;

- *Energie* : un nœud capteur est généralement muni d'une ou plusieurs piles normales pour alimenter tous ses composants, ce qui rend l'énergie la plus précieuse ressource dans un réseau de capteurs. De ce fait, la faible consommation d'énergie est une exigence principale pour les applications où une longue durée de vie du réseau est nécessaire ;
- *Puissance de calcul* : les réseaux de capteurs sont différents par rapport aux réseaux traditionnels. Parmi les points de différence nous pouvons citer la puissance de calcul. Les nœuds capteurs utilisent des microcontrôleurs de faibles fréquences. Par exemple le nœud populaire, MICA utilise un CPU Atmel de 7.37-MHz, 8-bit. Cette contrainte doit être considérée dans le développement d'applications pour les WSNs [48].

**Tableau 1. : Les différentes technologies des nœuds capteurs**

| | Modèle | Micro-Contrôleur | Type Radio | Radio (kbps) | RAM | Flash | EEPROM | Batterie |
|---|---|---|---|---|---|---|---|---|
| Crossbow Technology | MICA2 | Atmega128L MPR400 (8-bit) | ChipCon CC1000 | 38 | 4 KB | 128 KB | 4 KB | 2xAA |
| | MICAz | Atmega128L MPR2400 (8-bit) | ChipCon CC2420 | 250 | 4 KB | 128 KB | 4 KB | 2xAA |
| | TelosB | TI MSP 430 (16-bit) | ChipCon CC2420 | 250 | 10 KB | 48 KB | 16 KB | 2xAA |
| | Imote2 | Intel PXA271 (32-bit) | ChipCon CC2420 | 250 | 32 MB | 32 MB | | 3xAAA |
| Shockfish S A | TinyNode | TI MSP 430 (16-bit) | Semtech XE1205 | 153 | 10 KB | 48 KB | 16 KB | 2/3xAA |

### 1.3.7 MANETs versus WSNs

Bien qu'il y ait beaucoup de similitudes entre les MANETs et les WSNs, il existe aussi des différences essentielles entre ces deux types de réseaux ad hoc. Nous nous arrêtons sur ces points en parlant des spécificités qui leur sont communes, mais aussi de leurs différences.

Les MANETs et les WSNs partagent les mêmes problèmes de recherche [49], notamment : les caractéristiques temporelles variables des liaisons sans fil, la communication sans fil multi-sauts, l'alimentation électrique limitée, la possibilité de défaillances de liaison et le déploiement ad hoc de nœuds dans le réseau. Cependant, les MANETs diffèrent des WSNs de plusieurs façons. Tout d'abord, la destination dans les WSNs est connue (par exemple, station de base externe (SB)). Dans les WSNs, la communication est basée sur le plusieurs-à-un, alors que dans les réseaux MANET, elle est généralement basée sur le pair-à-pair. Deuxièmement, les données collectées par de nombreux capteurs dans les réseaux WSNs sont basées sur des phénomènes communs, il y a donc une forte

probabilité que ces données aient une certaine redondance. Troisièmement, les MANETs sont caractérisés par des topologies hautement dynamiques en raison de la mobilité des nœuds libres. Le tableau 1.3 montre une comparaison de haut niveau entre les MANETs et WSNs en mettant en évidence les principales différences.

**Tableau 1. : MANETs versus WSNs**

| MANETs | WSNs |
|---|---|
| Topologies dynamiques | Topologies statiques (dans la plupart des cas) |
| Les MANETs sont généralement des réseaux hétérogènes comportant des dizaines voir des centaines de nœuds | Les WSNs sont généralement des réseaux homogènes composés de milliers de nœuds et même et plus |
| La puissance peut être renouvelée ou rechargée | Besoin de fonctionner longtemps sur une minuscule batterie |
| Routage centré sur les adresses | Routage centré sur les données |
| Utilise la communication point à point | Utilise la communication point-multipoint |
| Optimise la QoS et les performances | Optimise l'utilisation de l'alimentation et non la qualité de service |

## Conclusion

Dans ce chapitre, nous avons présenté les spécificités des réseaux sans fil en général et celles des MANETs et des WSNs en particulier en insistant sur les aspects qui rendent l'application de la sécurité une tâche difficile. Bien que ces deux types de réseaux nécessitent des considérations de conception similaires, retenons tout de même que les MANETs présentent normalement des exigences moins strictes, car ils sont par définition des réseaux spontanés de courte durée avec des nœuds plus puissants. Les WSNs par contre ont des caractéristiques très particulières qui impliquent des considérations particulières qui ne sont pas communément présentes dans les MANETs.

# Chapitre 2 SECURITE ET GESTION DE CLES DANS LES MANETs ET LES WSNs

Avec le développement rapide de la technologie sans fil, les réseaux ad hoc ont émergé sous de nombreuses formes. Ces réseaux fonctionnent dans la bande de fréquences sans licence et ne nécessitent aucun investissement dans l'infrastructure, ce qui les rend attrayants pour les applications militaires, commerciales etc. Cependant, il existe de nombreux problèmes non résolus dans les réseaux ad hoc. Sécuriser de tels réseaux étant l'une des préoccupations majeures. Dans ce chapitre, nous donnons d'abord un aperçu sur la sécurité dans les MANETs et les WSNs avant d'introduire la question liée à la gestion des clés dans ces deux types de réseaux. Nous proposons également une vue d'ensemble sur la révocation de clés dans les MANETs et les WSNs, et montrons son importance dans le processus de sécurisation de ces réseaux via la gestion des clés.

## 2.1. Sécurité des MANETs et des WSNs

Les MANETs et les WSNs sont vulnérables aux attaques de sécurité car la transmission a lieu en milieu ouvert. Il n'y a pas de serveur centralisé, de station de surveillance ou d'administrateur, et les nœuds continuent à se joindre et à quitter le réseau. La détection de routes optimales pour l'économie d'énergie et de nœuds malveillants engendre des charges de communication [50].

Les réseaux câblés sont généralement sécurisés à l'aide de pare-feu et de dispositifs de chiffrement et surveillés à l'aide d'un proxy, d'un système de détection d'intrusion et de routeurs. Toutefois, ces périphériques ne sont pas disponibles dans le cadre des réseaux ad hoc tels que les MANETs et les WSNs. Les techniques de détection d'intrusion utilisées dans les réseaux filaires ne sont pas efficaces dans les réseaux ad hoc, car des nœuds inconnus peuvent rejoindre le réseau à un moment donné [50].

Les algorithmes cryptographiques asymétriques doivent être modifiés et personnalisés pour être utilisables dans le MANETs et les WSNs. L'infrastructure à clé publique (PKI) est difficile à mettre en œuvre avec un MANET car ils manquent généralement de connectivité avec Internet pour l'authentification à partir de l'autorité de certification (CA) [50]. La jonction de nouveaux nœuds à un emplacement distant rend également difficile la distribution de leur clés publiques aux autres nœuds du MANET à l'aide d'un canal sécurisé. Les algorithmes de sécurité pour les réseaux ad hoc sont conçus pour réduire l'exigence de calcul des nœuds ad hoc pour économiser de l'énergie. A ceci s'ajoute l'exigence primordiale de maintenir au minimum la dépendance de l'algorithme sur un nœud central. Les changements topologiques fréquents dans la conception du réseau, les nœuds actifs mais en hibernation et les nœuds compromis, rendent difficile la modélisation du réseau pour la simulation et l'analyse des protocoles de sécurité pour MANETs.

Dans la suite de cette section, les problèmes de sécurité sont analysés en référence, aux attaques, aux services et aux mécanismes de sécurité.

Les réseaux ad hoc fournissent généralement des services de sécurité et des mécanismes similaires à ceux déployés dans les réseaux locaux sans fil. Cependant, les caractéristiques représentatives d'un réseau ad hoc les rendent plus vulnérables [50]. Ces spécificités sont multiples. Elles peuvent être réparties en six grands thèmes [51] traitant des caractéristiques des nœuds, de la gestion de l'énergie, des caractéristiques du réseau, des technologies sans fil sous-jacentes, de la mobilité et de la configuration :

- **Caractéristiques des nœuds :**
  - Les nœuds formant un système hétérogène doivent s'interconnecter facilement ;
  - Certains nœuds du réseau peuvent avoir de faibles capacités de calcul [51].

- **Gestion de l'énergie :** L'énergie doit être conservée au maximum pour éviter d'incessantes recharges du système, qui diminuent sa mobilité. Les nœuds chercheront donc à se mettre en veille le plus souvent possible, ce qui provoquera alors une diminution de réactivité de l'ensemble du réseau [51].

- **Caractéristiques du réseau :**
  - La charge du réseau doit être distribuée équitablement entre les éléments en tenant compte de leur capacité respective [51] ;
  - Chaque élément d'un réseau ad hoc est autonome et possède à la fois les fonctionnalités de relais et de point de communication. L'administration de ces éléments reste interne au réseau ;
  - L'absence d'infrastructure centralisée sera une contrainte très forte pour la gestion des accès aux ressources du réseau [51].

- **Technologie sans fil :**
  - Les perturbations dues à l'environnement radio peuvent entraîner des diminutions de débit ;
  - Les réseaux sans fil ad hoc héritent de l'architecture propre aux technologies WLAN et WPAN, et notamment des couches physique et liaison de données de ces technologies [51].

- **Mobilité :**
  - Les éléments d'un réseau MANET étant fortement mobiles, leur sécurité physique est moins assurée que pour un poste de travail fixe, dans un bureau par exemple ;

- La topologie du réseau peut changer d'autant plus rapidement que les nœuds sont mobiles ;
- Des liens asymétriques peuvent se créer lorsqu'un élément muni d'un récepteur particulièrement sensible est capable de capter les émissions d'un autre nœud qui est hors de portée du premier élément [51].

- **Auto-configuration :**
  - L'auto configuration permet aux nœuds de s'intégrer facilement dans un réseau. Elle facilite la gestion du réseau, car l'interconnexion des éléments ne nécessite qu'un minimum d'intervention technique externe. Cette fonctionnalité est de plus en plus nécessaire pour un déploiement à grande échelle des réseaux ad hoc sans fil.

### 2.1.1 Les Attaques de Sécurité

Il y a beaucoup de types et de variétés d'attaques dans les MANETs et les WSNs. Toutes ces attaques peuvent être classées en fonction de différents aspects, à savoir la légitimité, l'interaction et la pile de protocoles de réseau [52].

#### 2.1.1.1 Classification basée sur la légitimité

Selon le statut légitime d'un nœud, une attaque se classifie en attaque externe ou interne. Les attaques externes sont effectuées par des nœuds qui ne sont pas membres légaux du réseau et les attaques internes proviennent d'un membre compromis à l'intérieur du réseau. Les attaques internes ne sont pas faciles à prévenir ou à détecter. Ces attaquants sont conscients des stratégies de sécurité et sont même protégés par elles. Comparées aux attaques externes, celles internes constituent une menace plus importante pour le réseau [53].

#### 2.1.1.2 Classification basée sur l'interaction

En termes d'interaction, une attaque est divisée en attaque passive et active. Les attaques passives ne perturbent pas la communication. Au lieu de cela, elles interceptent et capturent les paquets pour lire l'information. La détection des attaques passives est compliquée, puisque l'opération réseau n'est pas effectuée. L'une des solutions utilise des méthodes de chiffrement. Ce mécanisme cryptera les données pendant le processus de transmission. Il protège les données contre les indiscrètes en rendant difficile l'accès à information pendant le processus de communication [53].

Une attaque active fait référence aux attaques qui tentent de modifier, injecter, supprimer ou détruire les données échangées sur le réseau. Ces attaques peuvent être exécutées par des attaquants internes ou externes, le tableau 2.1 explique le type et l'exemple de l'attaque basée sur l'interaction.

**Tableau 2. : Type d'attaque basée sur l'interaction [53]**

| Type of attack | Example of attack |
|---|---|
| Passive attack | Eavesdropping, traffic analysis |
| Active attack | Impersonation, Repudiation, Routing Attacks, black hole, neighbor attack, wormhole, denial of service (DoS), information or location disclosure, rushing attack, jellyfish attack, malign attack, partition attack, detour attack, routing table poisoning, packet replication, session hijacking . |

#### 2.1.1.3 Classification d'attaques par pile de protocole réseau

Les attaques peuvent également être classées en fonction de la couche ciblée dans la pile de protocoles. Le tableau 2.2 montre la cartographie des attaques dans chaque couche [54].

**Tableau 2. : Attaque au niveau de chaque couche [54].**

| Layer | Security Issues |
|---|---|
| Application | Repudiation, data corruption, virus, backdoor. |
| Transport | Session hijacking, SYN flooding |
| Network | Wormhole, blackhole, byzantine, flooding, resource consumption, location disclosure attacks |
| Data link layer | Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness |
| Physical | Preventing signal jamming, interceptions, eavesdropping |
| Multi-layer attacks | DoS, Impersonation, replay, man-in-the-middle. |

Les MANETs et les WSNs peuvent faire l'objet d'un grand nombre d'attaques, chacune avec ses objectifs propres. Par exemple, certaines attaques visent à affecter l'intégrité des messages qui transitent dans le réseau, tandis que d'autres visent à réduire la disponibilité du réseau ou de ses composants. Les attaques se produisent souvent par l'insertion d'éléments intrus dans le réseau. Il existe aussi des attaques contre l'environnement extérieur au réseau, lesquelles provoquent des altérations ou des interférences sur les signaux transmis. Une bonne classification des attaques est présentée dans [55]. Les attaques les plus connues dans les WSNs [56], mais également dans les MANETs sont :

- **Ecoute du réseau (*eavesdropping*) :** Du fait que les transmissions se font en diffusion par les ondes radio, aucun contrôle d'accès au signal n'est possible, ce qui est d'autant plus vrai que le réseau peut être déployé dans un environnement ouvert accessible à tout le monde. Il est donc très facile d'intercepter des données échangées sur un réseau de capteurs et d'accéder à leur contenu si aucun service de confidentialité n'est prévu ;

- **Attaque physique (*tampering*) :** Comme les WSNs sont très souvent déployés dans des zones sans aucune protection, ils sont très exposés aux attaques physiques qui peuvent être considérées sous différents points de vue. L'un est lié au matériel qui n'est pas qualifié d'inviolable. Dans ces conditions, une attaque aura pour but de récupérer des outils cryptographiques comme les clés utilisées pour le chiffrement. Un autre objectif serait de reprogrammer le capteur pour perturber le réseau et l'application en provoquant volontairement un comportement anormal du nœud. La seconde attaque physique consisterait simplement à supprimer le capteur du réseau en le détruisant (on retombe sur la question de l'inviolabilité) ou en le subtilisant [57] ;
- **Attaque de l'identité multiple (*sybil attack*) :** Avec cette attaque, un nœud malveillant peut revendiquer différentes identités afin de participer à des algorithmes distribués, tels que l'élection et de prendre de l'avantage sur les nœuds légitimes. Un nœud malveillant peut être capable de déterminer le résultat de n'importe quel vote en faisant voter toutes ses identités multiples pour une même entité. Les techniques d'authentification et de chiffrement peuvent empêcher un étranger de lancer une attaque sybil sur le réseau de capteur [56] ;
- **Attaque du trou noir (*blackhole ou sinkhole*) :** Un nœud falsifie les informations de routage pour forcer le passage des données par lui-même. Sa seule mission est ensuite de ne rien transférer, créant ainsi une sorte de puits ou trou noir dans le réseau. L'intrus (nœud malveillant, qui s'introduit illégitimement) se place à un endroit stratégique de routage dans le réseau et supprime tous les messages qu'il devrait retransmettre, causant la suspension du service de routage du réseau pour les routes qui passent par le nœud intrus [56] ;
- **Attaque du trou gris (*greyhole*) :** Une variante de l'attaque précédente est appelée trou gris, dans laquelle seuls certains types de paquets sont ignorés par le nœud malicieux. Par exemple, les paquets de données ne sont pas retransmis alors que les paquets de routage le sont [56] ;
- **Brouillage radio (*jamming*) :** L'intrus inonde avec du bruit les fréquences radio utilisées par le réseau de manière à empêcher les transmissions et/ou les réceptions de messages. Ce type d'attaque peut affecter tout ou une partie du réseau selon la portée radio de l'intrus. Dans ce cas-là, l'intention est de provoquer un déni de service [56] ;
- **Relais sélectif (*selective forwarding*) :** L'intrus néglige son rôle de routeur et ne transmet pas certains messages, qui sont choisis selon certains critères ou même aléatoirement [56] ;
- **Attaque du trou de ver (*wormhole*) :** L'intrus capture un message et, en utilisant un canal de faible latence, le retransmet vers un lieu distant dans le réseau. Le canal ainsi créé fait transiter un message à un endroit du WSN auquel il ne devrait normalement pas arriver. Cette attaque a une influence notable sur le routage dans le réseau [56] ;
- **Rejeu, Délai et Altération de Données :** L'intrus répète, retarde ou altère le contenu des messages en transit. Les messages peuvent contenir des données de perception prélevées et des données de configuration ou de routage. Ces types d'attaques visent, entre autres, à créer

des boucles, attirer ou éloigner du trafic, augmenter ou diminuer le nombre de routes, générer de fausses erreurs, partitionner le réseau, et augmenter la latence de distribution des données [56];

- **Attaque par chantage (*Blackmailattack*) :** Un nœud malicieux fait annoncer qu'un autre nœud légitime est malicieux pour éliminer ce dernier du réseau. Si le nœud malicieux arrive à attaquer un nombre important de nœuds, il pourra perturber le fonctionnement du réseau [56] ;
- **Attaque par l'inondation de "*HELLO*" :** De nombreux protocoles de routage utilisent des paquets "HELLO" pour découvrir les nœuds voisins et ainsi établir une topologie du réseau. L'attaque la plus simple consiste pour un attaquant à envoyer un flot de messages Hello pour inonder le réseau et empêcher d'autres messages d'être échangés [56] ;
- **Epuisement de la batterie (exhaustion de batterie) :** Cette attaque de déni de service est redoutable car elle vise à épuiser les batteries des nœuds composants le réseau de manière à réduire la durée de vie du réseau. Elle peut consister à injecter de nombreux messages dans le réseau, qui conduisent les nœuds à gaspiller leur énergie en retransmissions inutiles [56].

### 2.1.2 Les Services de Sécurité

Ils sont conçus pour contrer les attaques de sécurité en utilisant un ou plusieurs mécanismes de sécurité [58]. La plupart de ces services, tels que la confidentialité, l'authentification, l'intégrité, la non-répudiation, le contrôle d'accès, la disponibilité, et le respect de la vie privée dérivent de principaux buts de sécurité de tout système.

**Authentification**

L'authentification est la vérification des revendications concernant l'identité d'une source d'information. L'authenticité est essentiellement l'assurance que les participants à la communication sont authentiques et non imitateurs. Il est nécessaire que les participants à la communication prouvent leurs identités comme ce qu'ils ont prétendu, en utilisant certaines techniques, afin d'assurer l'authenticité. L'authentification peut être fournie en utilisant le chiffrement avec la fonction de hachage cryptographique, la signature numérique et les certificats. Sans authentification, un adversaire pourrait se faire passer pour un nœud, obtenant ainsi un accès non autorisé à des ressources et des informations sensibles et interférant avec les opérations des autres nœuds [59].

**Confidentialité**

La confidentialité signifie que seules les personnes ou les systèmes autorisés peuvent lire ou exécuter des données ou des programmes protégés. Il convient de noter que la sensibilité de l'information dans les MANETs peut se détériorer beaucoup plus rapidement que dans d'autres types de réseaux.

**Intégrité**

L'intégrité signifie que l'information n'est pas modifiée ou corrompue par des utilisateurs non autorisés ou par l'environnement. L'intégrité garantit l'identité des messages lorsqu'ils sont transmis. L'intégrité peut être compromise principalement de deux façons : altération malveillante et altération accidentelle.

**Disponibilité**

La disponibilité fait référence à la capacité du réseau à fournir des services selon les besoins. Le terme Disponibilité signifie qu'un nœud doit conserver sa capacité à fournir tous les services conçus, quel que soit l'état de sécurité de celui-ci.

**Non-Répudiation**

La non-répudiation garantit que les actions engagées ne peuvent pas être refusées. Dans les MANETs et les WSNs, les objectifs de sécurité d'un système peuvent changer selon différents modes (par exemple, temps de paix, transition vers la guerre et temps de guerre d'un réseau militaire). Les caractéristiques de ces réseaux les rendent vulnérables à de nombreuses nouvelles attaques.

**Contrôle d'accès / Autorisation**

L'autorisation est un processus dans lequel une entité reçoit un droit d'accès qui spécifie les privilèges et autorisations dont elle dispose, et ne peut pas être falsifiée par l'autorité de certification. L'autorisation est généralement utilisée pour attribuer différents droits d'accès à différents niveaux d'utilisateurs.

**Anonymat**

L'anonymat signifie que toutes les informations qui peuvent être utilisées pour identifier le propriétaire ou l'utilisateur actuel du nœud doivent par défaut être gardées privées et ne pas être distribuées par le nœud lui-même ou le logiciel du système. Ce critère étant étroitement lié à la préservation de la vie privée, nous devrions essayer de protéger la confidentialité des nœuds contre la divulgation arbitraire à d'autres entités.

**Extensibilité / Scalabilité**

L'extensibilité n'est pas directement liée à la sécurité, mais c'est une question très importante, qui a un grand impact sur les services de sécurité. Les mécanismes de sécurité doivent être évolutifs pour pouvoir gérer les variations de la taille du réseau causées par l'arrivé ou le départ de nœuds.

En ce qui concerne la fourniture des services de sécurité, Il convient de noter que :

au sein d'un réseau sans fil ad hoc, coopérer présente un risque s'il n'y a aucun contrôle des participants. L'authentification des parties apparait donc comme la pierre angulaire d'un réseau sans fil ad hoc sécurisé [60]. En effet, comment assurer une quelconque confidentialité et intégrité des messages échangés si, dès le départ, on n'est pas sûr de communiquer avec la bonne entité? Contrairement au réseau filaire, il n'est pas nécessaire de pénétrer dans un local physique pour accéder au réseau. Si l'authentification est mal gérée, un attaquant peut s'attacher au réseau sans fil et injecter des messages erronés. L'intégrité des messages échangés est donc une exigence importante pour ces réseaux. L'intégrité des nœuds est, elle aussi, primordiale car les éléments d'un réseau ad hoc sont moins sujets à surveillance. En effet, ils ne sont pas confinés dans un bureau. Un attaquant peut subtiliser un appareil, le corrompre avant de le restituer discrètement. Une fois les parties authentifiées, la confidentialité reste un point important étant donné que les communications transitent via ondes radio, et sont donc potentiellement accessibles à tout possesseur du récepteur adéquat ;

La disponibilité est une propriété difficile à gérer dans les réseaux sans fil ad hoc, vu les contraintes qui pèsent sur ces réseaux, notamment une topologie dynamique, des contraintes en ressources des nœuds, surtout ceux d'un réseau de capteurs, les liaisons sans fil pouvant être facilement brouillées ou perturbées [60] ;

Certaines applications peuvent nécessiter la discrétion sur l'identité des participants qui collaborent au réseau sans fil ad hoc : par exemple un vote anonyme au cours d'une conférence. De plus, les différents gadgets électroniques qui formeront les nœuds des réseaux ad hoc, ont en toute probabilité, la possibilité de garder la trace des préférences afin de faciliter le quotidien et d'offrir des services toujours plus appropriés. Cette tendance va pourtant à l'encontre de la protection de la vie privée de tout un chacun [60].

### 2.1.3 Mécanismes de sécurité

La plupart des services de sécurité précédemment mentionnés, notamment la confidentialité, l'intégrité et l'authentification, peut être assurée en utilisant différents mécanismes cryptographiques ou non. La cryptographie est l'ensemble des méthodes permettant de sécuriser le transfert et le stockage des données de telle sorte que seulement les nœuds concernés puissent retrouver les données initiales [61].

Un message est transformé pour le rendre incompréhensible pour toute personne non autorisée et empêcher toute modification. Le destinataire peut récupérer les données en clair et s'assurer que celles-ci n'ont pas été altérées. Les transformations en question sont basées sur des fonctions mathématiques appelées algorithmes cryptographiques.

Les algorithmes cryptographiques de chiffrement sont constitués de deux opérations symétriques :

– Le chiffrement : opération permettant de transformer un message en clair en un message chiffré (incompréhensible) ;

– Le déchiffrement : opération symétrique à celle de chiffrement permettant de retrouver le message en clair à partir du chiffré.

Puisque les algorithmes de chiffrement/déchiffrement sont publics (donc à priori connus de tous), le secret repose exclusivement sur le paramètre clé [62]. Ainsi, les entités capables de déchiffrer les messages connaissent une information supplémentaire capitale dans ce processus.

Nous distinguons deux types d'algorithmes cryptographiques réversibles : ceux utilisant la même clé (symétrique) et ceux utilisant deux clés différentes (asymétrique).

#### 2.1.3.1 Cryptographie symétrique

Les mécanismes de chiffrement symétriques à clé secrète, permettent à des entités (généralement deux) d'effectuer des opérations de chiffrement/déchiffrement en utilisant une seule clé. En ce sens, toute personne qui peut chiffrer des messages, et donc qui connait la clé secrète, peut également déchiffrer. Ce mode de chiffrement se comporte comme une boite fermée avec une serrure où les entités désirant communiquer doivent posséder chacun une clé leur permettant d'ouvrir et fermer la boite.

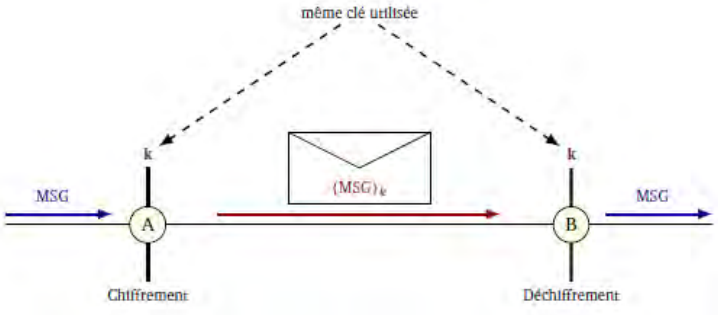



**Figure 2. : Chiffrement symétrique**

Dans la figure 2.1 ci-dessus, le nœud *A* désire envoyer un message à *B*. Il chiffre donc son message avec la clé *k*. En recevant le message, *B* déchiffre en utilisant la même clé *k*. Il est important de noter qu'il y a une phase initiale ou les deux nœuds se mettent d'accord sur la clé secrète qui sera utilisée.
Ce mode de protection de la confidentialité des données présente l'avantage d'être peu gourmand en termes de puissance de calcul. Cependant, son principal inconvénient réside dans la distribution de la clé secrète avant de commencer la communication. Cette phase d'échange de clés est d'autant plus importante lorsque les entités désirant communiquer de manière sécurisée en point à point sont de plus

en plus nombreuses. Parmi les algorithmes de chiffrement symétrique les plus connus, nous citons AES [63] (*Advanced Encryption Standard*) ou encore DES [64] (*Data Encryption Standard*).

### 2.1.3.2 Cryptographie asymétrique

On peut assimiler les mécanismes à clés publique à un coffre-fort dont une seule personne possède la clé. Ce coffre est laissé ouvert à la disposition de toute personne désirant lui envoyer un message. Une fois fermé, seul le propriétaire du coffre pourra l'ouvrir.

En pratique, il s'agit d'une fonction de chiffrement à deux clés : une clé publique, qui est fournie à tous ceux désirant envoyer un message confidentiel à une entité, et une clé privée qui sert à déchiffrer les messages qui lui sont destinés. Ces deux clés sont liées de telle sorte que seule la clé privée peut déchiffrer un message chiffré par la clé publique correspondante.

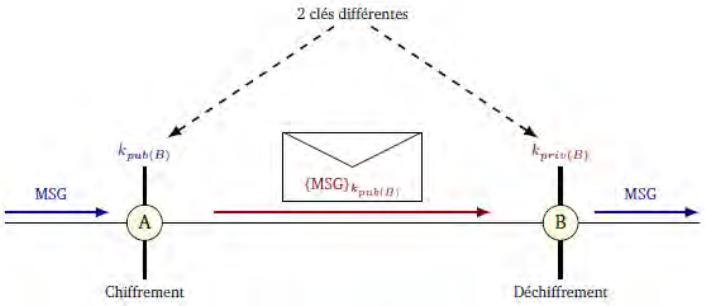



**Figure 2. : Chiffrement asymétrique**

Dans la figure 2.2, le nœud *A* désire transmettre le message *MSG* à *B*. Il récupère alors la clé publique de la destination $k_{pub}(B)$ et chiffre le message en utilisant cette clé ($\{MSG\}k_{pub}(B)$). Le message chiffré est envoyé sur le canal de communication. La destination *B* déchiffre le message reçu en utilisant sa clé secrète $k_{priv}(B)$ pour obtenir le message en clair envoyé par *A*.

Un des algorithmes à clé publique les plus populaires est RSA [65] du nom de ses concepteurs *Rivest*, *Shamir* et *Adleman*. Cet algorithme est basé sur le problème de factorisation d'entiers. La cryptographie asymétrique nécessite beaucoup plus de puissance de calcul par rapport à la cryptographie symétrique. C'est pourquoi, elle est principalement utilisée dans la phase d'échange de clé de session symétrique entre les entités, clé qui est utilisée par la suite pour chiffrer les communications.

En termes d'authenticité, la cryptographie asymétrique est la seule à garantir véritablement l'authentification de la source puisque la cryptographie symétrique ne distingue pas la source de la destination.

#### 2.1.3.3 Fonctions de hachage cryptographiques

Pour garantir l'intégrité des messages, une fonction de hachage cryptographique peut être utilisée. C'est une fonction non réversible (*one-way*) pour produire un condensé (aussi appelé empreinte ou haché ou digeste) ayant les propriétés suivantes :

- Unique et de taille fixe ;
- Il est impossible (techniquement parlant et dans un temps raisonnable) de retrouver le message d'origine à partir du condensé ;
- Sachant un message donné et son empreinte, il est très difficile en utilisant une fonction de hachage de générer un autre message qui donne la même empreinte (*weak collision resistance*) ;
- Il est impossible de trouver deux messages produisant le même condensé (*strong collision resistance*).

Le temps négligeable pour le calcul du condensé constitue un avantage pour l'utilisation des fonctions de hachage cryptographiques. Ces fonctions sont utilisées entre autres pour la signature numérique (le condensé du message est calculé et lui seul est signé) et aussi pour des mécanismes d'authentification par mot de passe sans stockage de ce dernier.

La figure 2.3 montre l'utilisation de la fonction de hachage pour préserver l'intégrité des messages échangés entre deux entités (*A* et *B*).

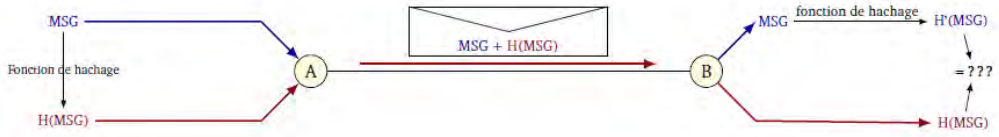



**Figure 2. : Utilisation des fonctions de hachage cryptographiques**

#### 2.1.3.4 Message Authentication Code

L'algorithme MAC (*Message Authentication Code*) est un cas particulier de fonction de hachage. Comme une fonction de hachage, l'algorithme MAC produit un condensé de taille fixe (appelé code d'authentification de message - MAC). Mais à la différence de celui-ci, il utilise en plus une clé secrète.

Le MAC peut être utilisé pour vérifier simultanément l'intégrité de données comme toute autre fonction de hachage et l'authenticité de la source des données.

L'émetteur calcule le MAC en utilisant une fonction de hachage qui prend en paramètre la symétrique *k*. Ensuite, le message et le MAC sont transmis au destinataire. Celui-ci calcule le MAC du message reçu en appliquant la même fonction de hachage avec la clé symétrique *k* et compare le résultat obtenu avec le MAC reçu. S'il y a égalité, le message n'a pas été altéré et il provient bien du nœud supposé (voir figure 2.4).

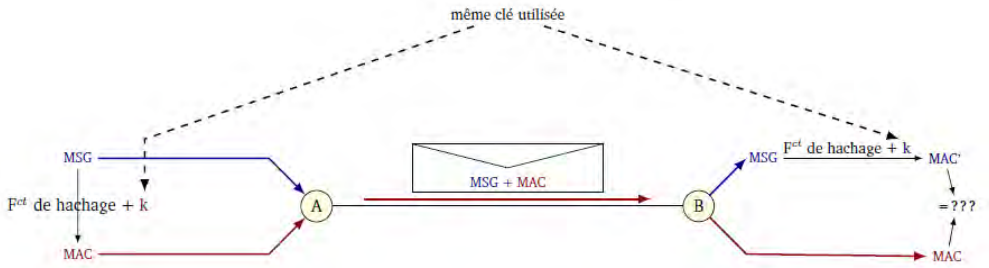



**Figure 2. : Utilisation des fonctions de hachage à clé symétrique**

Dans la pratique, une méthode de calcul du MAC à base de fonction de hachage plus élaborée et plus sûre est utilisée : il s'agit du HMAC (*keyed-Hash Message Authentication Code*) [66] qui a fait l'objet de la RFC 2104.

### 2.1.3.5 Signature numérique

La signature numérique est un mécanisme assurant l'intégrité d'un message ainsi que l'authentification de la source de celui-ci. Ceci est possible grâce à la cryptographie asymétrique.
Lorsqu'une source désire transmettre des données à une destination, elle applique la fonction de hachage sur le message à envoyer et le condensé obtenu est chiffré avec sa clé privée. Le résultat obtenu constitue la signature et est envoyé avec le message (voir figure 2.5). La destination utilise la clé publique de la source pour déchiffrer le condensé et le compare au condensé obtenu en appliquant la fonction de hachage sur le message reçu. Ainsi, la destination garantit que le message vient de la source et non pas d'un nœud usurpant son identité, et garantit aussi que le message n'a pas été altéré lors du transfert.

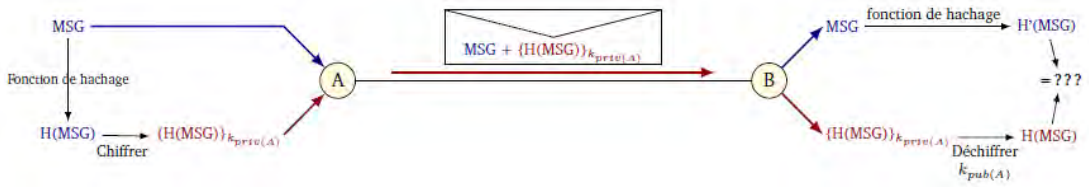



**Figure 2. : Signature numérique**

#### 2.1.3.6 Certificat électronique

Un certificat électronique est une carte d'identité numérique liant une entité physique à une entité numérique. Une autorité de certification (*Certification Authority*, CA) fait office de tiers de confiance et atteste le lien. Ainsi, une entité peut ajouter son certificat lors de la signature du message avant de l'envoyer. Ce certificat contiendra entre autres sa clé publique. Le standard le plus utilisé pour la création des certificats numériques est le X.509 [61].

Dans la section suivante, nous donnons un aperçu sur la gestion des clés dans les MANETs et les WSNs et présentons brièvement les solutions proposées jusqu'ici dans la littérature.

## 2.2. La Gestion des clés

Les nœuds dans un réseau doivent être en mesure de communiquer de manière sécurisée les uns avec les autres. Pour sécuriser la communication entre deux entités, ces dernières doivent avoir une valeur secrète, ou clé. Pour rendre possible cette communication sécurisée, il est nécessaire, pour les nœuds, d'avoir accès aux clés cryptographiques.

La gestion des clés dans les réseaux a une importance cruciale. En cas d'utilisation de systèmes cryptographiques, comme le chiffrement ou les signatures numériques pour protéger à la fois le contrôle et le trafic de données, un service de gestion de clés est toujours requis. Les différentes manières de sécuriser les communications concernent les entités partageant une clé ou les entités possédant des clés différentes. La gestion des clés est le processus par lequel les clés sont distribuées aux nœuds dans le réseau, et la façon dont elles sont si nécessaire mises à jour, effacées, révoquées ou stockées.

La gestion des clés peut concerner plusieurs étapes, s'agissant de systèmes à clés symétrique ou asymétrique.

Il s'agit notamment de [67] :

1. l'initialisation des utilisateurs du système ;
2. la création, la distribution et l'installation des clés cryptographiques ;
3. l'organisation de l'utilisation des clés cryptographiques ;
4. la mise à jour, la révocation et la destruction des clés cryptographiques ;
5. l'archivage des clés cryptographiques.

Le but de la première étape est l'amorçage du système. Cela pourrait inclure diverses autres opérations non cryptographiques, telles que la vérification de l'information sur les utilisateurs, la fourniture d'identités aux utilisateurs du système, la vérification des logiciels nécessaires pour participer au processus de gestion des clés, etc. Cette partie est suivie de la création et la distribution

des clés cryptographiques, qui peuvent se faire de manière centralisée ou décentralisée. Les clés qui ont été distribuées sont alors installées sur les différents nœuds. Lors de la troisième phase les clés cryptographiques sont utilisées pour protéger la communication entre les différents nœuds. Cela se fait en utilisant des clés pour chiffrer les données et contrôler le trafic échangé entre les différents nœuds du réseau. La quatrième étape est cruciale pour faire face à diverses menaces qui pourraient aboutir à une compromission des clés. Cela pourrait conduire à un manque de confidentialité ou d'authenticité, ainsi qu'à l'utilisation non autorisée des clés. Dans ce cas, des processus de gestion doivent veiller à ce que les clés compromises soient annulées (révoquées). En outre, il peut parfois être nécessaire de remplacer les clés compromises. Bien sûr, le remplacement des clés compromises n'est pas nécessaire pour les nœuds sous le contrôle de l'adversaire. Enfin, la cinquième étape pourrait être nécessaire dans le cas où les clés cryptographiques sont à sauvegarder.

La majorité des mécanismes utilisés pour fournir les services de sécurité nécessite l'utilisation de clés cryptographiques, qui doivent être partagées entre les parties communicantes. Les processus de gestion de clés impliquent des techniques différentes lors de l'examen des systèmes cryptographiques symétriques et asymétriques, des réseaux filaires ou sans fil. L'approche typique de gestion des clés dans les réseaux traditionnels filaires est basée sur l'utilisation d'un tiers de confiance (*Trusted Third Party*, TTP) et d'une Infrastructure à clés publiques (*Public Key Infrastructure*, PKI).

### 2.2.1 Trusted Third Parties (TTPs)

Nous faisons la distinction entre TTP externe, interne, central et distribué [3]. Un TTP externe ou *off-line* TTP est une entité en dehors du réseau, qui a la confiance de tous les nœuds du réseau. Le TTP peut consister en une seule entité centrale ou $n$ entités distribuées. La dernière mise en œuvre est parfois choisie afin d'accroître la confiance au TTP, car la confiance peut être maintenue, même si certains des TTPs peuvent ne pas être dignes de confiance. Un TTP interne est un TTP distribué composé de $N$ nœuds de réseau, où $N \leq n$, $n$ étant le nombre total de nœuds dans le réseau. Ici, le pouvoir et les capacités d'un TTP sont distribués à $N$ nœuds du réseau. La distribution du pouvoir est souhaitable afin d'éviter un unique point de défaillance. Cela est nécessaire dans les MANETs à cause de la probabilité de compromissions de nœuds. Il y'a deux cas possibles de TTPs internes distribués : TTP distribué avec des nœuds spéciaux et TTP distribué avec des nœuds réseau conventionnels.

Dans le premier cas $N$ nœuds spéciaux, qui sont plus puissants que les autres $n - N$ nœuds du réseau en termes de calcul et de capacité de la batterie, représentent le TTP. Ces nœuds ont été initialisés au cours de la phase d'initialisation du réseau. Toutefois, cette approche est en contradiction avec l'hypothèse que les nœuds d'un MANET présentent des contraintes similaires.

Dans le second modèle, tout groupe de $N$ nœuds de réseau peut être choisi pour représenter le TTP. Notons que, tout en permettant plus de fonctionnalités et de flexibilité, l'usage de TTPs distribués on-line présente des inconvénients car, imposent toujours un lot supplémentaire de calcul et un coût de communication au réseau en raison de l'usage de systèmes de seuil.

Un TTP peut avoir plusieurs rôles dans un réseau, par exemple, un TTP peut initialiser les nœuds avec les clés nécessaires au cours de l'initialisation du réseau ou avant qu'ils ne rejoignent le réseau, distribuer des clés de session aux nœuds qui souhaitent communiquer de manière sécurisée les uns avec les autres durant le fonctionnement du réseau, ou aider à vérifier les certificats de clés publiques en fournissant des listes de révocation de certificats.

*Key Distribution Center* (KDC), *Key Translation Center* (KTC) et *Certificate Authorities* (CA) sont des exemples de TTP. KDC et KTC sont destinés aux systèmes à clés symétriques et CA est destiné aux systèmes à clés publiques. KDC et KTC sont utilisés pour simplifier la gestion des clés. Au lieu que chaque utilisateur ait à partager une clé secrète avec chaque autre utilisateur, il lui suffit d'en partager une avec le TTP. Cela réduit le nombre de clés à gérer de $n\ (n-1)\ /\ 2$ à $n$, où $n$ est le nombre total d'utilisateurs.

### 2.2.2 Public Key Infrastructure (PKI)

L'utilisation de la cryptographie à clé publique nécessite que l'authenticité de la clé publique puisse être établie. Une approche simple exige que deux utilisateurs qui souhaitent communiquer échangent leurs clés publiques de manière authentifiée. Cela demanderait la distribution initiale de $n\ (n-1)$ clés publiques [1]. Cependant, par le biais d'un TTP, qui distribue des certificats, seule sa clé publique doit être distribuée à chacun des utilisateurs.

Une PKI fournit les mécanismes nécessaires à la gestion des certificats et comprend les éléments illustrés à la Figure 2.6.

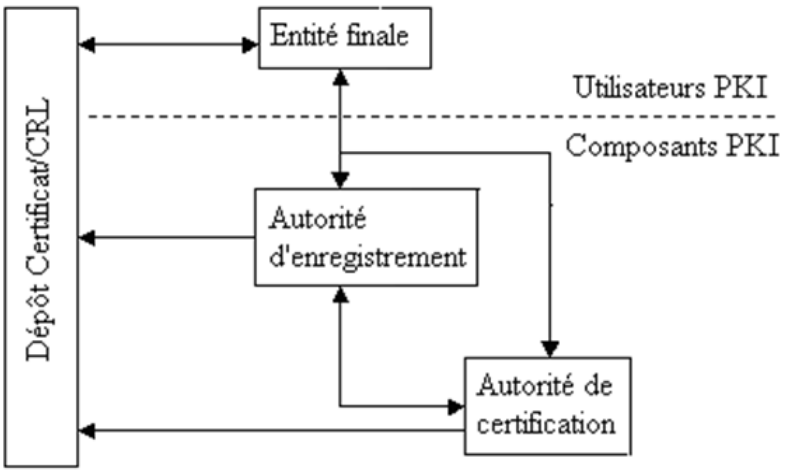


**Figure 2. : Principaux composants d'une PKI.**

Une entité finale est soit un utilisateur de certificats, soit un sujet auquel un certificat a été délivré. L'autorité de certification (CA) est le composant responsable de la délivrance et de la révocation des certificats, alors que l'autorité d'enregistrement (RA) est chargée d'établir l'identité du sujet du certificat et la correspondance entre lui et sa clé publique. Les fonctions d'enregistrement peuvent être exécutées par la CA, et donc la RA est une composante optionnelle. Les services de base suivants devraient être fournis par les composants PKI décrits ci-dessus :

- Inscription ;
- Initialisation ;
- Certification ;
- Mise à jour de clés ;
- Révocation ;
- Avis de révocation et de distribution de certificats.

D'autres services, qui peuvent être fournis par la PKI, comprennent la récupération de clés, la génération de clés, la non répudiation etc.

## 2.3. Gestion des clés dans les MANETs et les WSNs

### 2.3.1 Contraintes de conception

La gestion des clés dans les réseaux ad hoc est plus difficile comparée à la gestion des clés dans les réseaux traditionnels filaires. Dans [1] il est montré que cette difficulté est principalement liée à deux facteurs notamment :

- les exigences en infrastructure des solutions traditionnelles ;
- la facilité de mise en œuvre des attaques de sécurité dans les MANETs.

En se basant sur l'analyse faite dans [68] nous ajoutons alors à ces deux facteurs :

- les fortes contraintes en ressource des nœuds (cas des réseaux de capteurs sans fil).

La figure 2.7 suivante résume les contraintes, découlant des propriétés des MANETs et des WSNs, à prendre en compte lors de la conception d'une solution de gestion de clés.

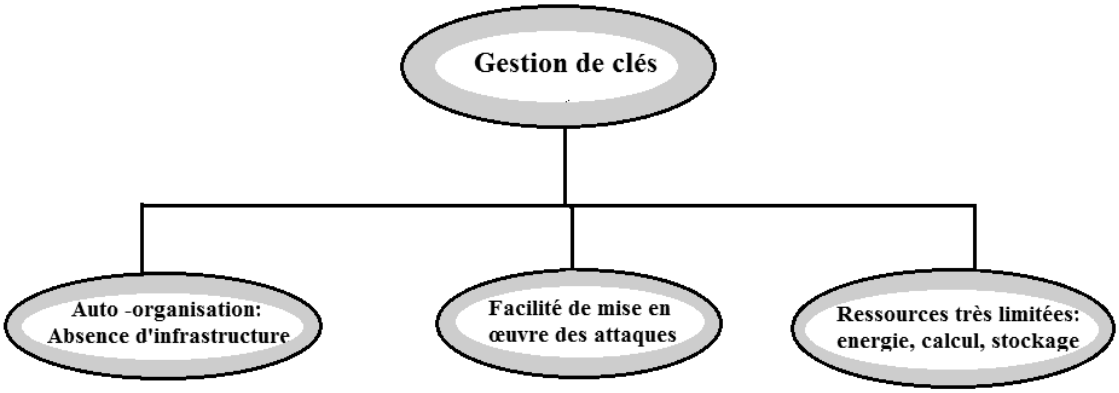



**Figure 2. : Contraintes de conception de solutions de gestion de clés**

Les solutions traditionnelles de gestion des clés présentent des exigences en infrastructure qui les rendent inappropriées aux réseaux ad hoc. Ces exigences se réfèrent à trois types principaux d'infrastructures [1] notamment :

- L'infrastructure de routage : il s'agit de routeurs et liaisons de communication stables, interconnectant les nœuds du réseau.
- L'infrastructure serveur : regroupant les serveurs *on-line* tels que les serveurs DNS, DHCP et CA fournissant des services aux nœuds du réseau.
- L'infrastructure administrative : assurant l'enregistrement des utilisateurs du réseau, l'émission de certificats et la gestion des autres tâches de configuration du réseau.

L'utilisation d'un TTP pour fournir les services de gestion des clés implique la présence d'une infrastructure administrative. Une telle infrastructure peut être présente dans certaines applications de réseau ad hoc, par exemple les réseaux militaires ou les réseaux de capteurs, mais pas dans de nombreux cas, par exemple les applications de réseaux collaboratifs. Même en cas de présence d'une infrastructure administrative, les solutions qui sont basées sur l'usage d'un TTP rencontrent des problèmes. Un des problèmes les plus évidents est qu'elles dépendent de la disponibilité d'un serveur central. Cependant, en raison de l'absence d'infrastructure de routage et d'une topologie hautement dynamique dans le cas des MANETs, une telle disponibilité ne peut pas être assurée. Enfin l'existence d'un serveur ne peut pas être assurée dans certaines applications MANET. Par exemple dans des réseaux ad hoc spontanés formés lorsque deux ou plusieurs PDA se trouvent à portée l'un de l'autre.

Une des menaces les plus évidentes dans les MANETs et les WSNs est celle de l'écoute clandestine, puisque toutes les communications sont réalisées via des canaux sans fil. Toutefois une menace plus sérieuse, et qui complique davantage le problème de l'accord de clés, est celle des attaques actives. En raison de l'absence d'infrastructure de routage, les nœuds participent à la

retransmission des messages pour le compte d'autres nœuds. Une conséquence directe en est qu'un attaquant peut facilement modifier les messages et exécuter une attaque man-in-the-middle.

Puisque les nœuds des réseaux ad hoc sont plus exposés que les nœuds d'un réseau traditionnel, le risque de compromission doit généralement être considéré comme supérieur. Cela implique que l'approche directe, améliorant la disponibilité d'un TTP en le distribuant à de multiples nœuds, est inappropriée. La raison est que cela affaiblit la sécurité, puisqu'elle expose plus les TTPs aux attaques. Si l'un de ces TTPs est compromis, la sécurité de l'ensemble du réseau l'est aussi. Dans le cas où le TTP est une CA une approche plus fiable est d'utiliser un système de partage secret à seuil pour partager le secret entre un nombre de nœuds dans le réseau. La compromission d'au plus $k-1$, $k$ étant le seuil du partage secret, laisse toujours la sécurité du service intacte [1].

Les menaces mentionnées ci-dessus ne sont pas seulement présentes dans les MANETs et les WSNs, elles existent aussi dans les réseaux traditionnels. Toutefois l'effort requis par l'adversaire pour parvenir à ses fins est considérablement plus réduit dans ces réseaux que dans les réseaux traditionnels. Ainsi les solutions qui ont été proposées pour les réseaux filaires ne sont pas toujours applicables aux réseaux ad hoc.

PKI a été mis en place pour l'infrastructure Internet filaire. Avec l'avènement des réseaux sans fil, en général, et des MANETs, en particulier, la même a continué à être utilisé sans aucun changement dans l'infrastructure ou le protocole. Ceci est particulièrement dû aux avantages principaux qu'offre l'usage des algorithmes à clés publiques, qui sont la facilité de gestion des clés et leur fiabilité. Cependant l'adoption de ces algorithmes présente aussi des inconvénients, à savoir la consommation d'énergie due au calcul des algorithmes à clé publique, la consommation d'énergie due à la transmission des certificats, et le stockage des clés connues pour être plus grandes que les clés symétriques.

Pour adresser ces limites, les WSNs emploient majoritairement des mécanismes à clés symétriques pour l'établissement de la confiance entre les nœuds à un coût plus réduit de la consommation d'énergie et d'espace de stockage. Cependant avec cette dernière approche de gestion des clés basée sur les clés symétriques, l'échange de clés est beaucoup plus compliqué et habituellement, une seule clé symétrique est utilisée entre deux parties communicantes, sur une seule session ou sur une période limitée.

### 2.3.2 Les Approches de Gestion des clés dans les MANETs et les WSNs

Les protocoles qui dépendent de l'utilisation d'un TTP ne sont pas appropriés au scénario des réseaux ad hoc. La raison principale est l'absence d'infrastructure fiable [34]. Cependant quelques propositions basées sur la modification de cette approche ont été faites. Les auteurs de [69] classifient

ces propositions en tant que schémas de gestion de clés symétriques, asymétriques et hybrides comme illustré sur la figure 2.8.

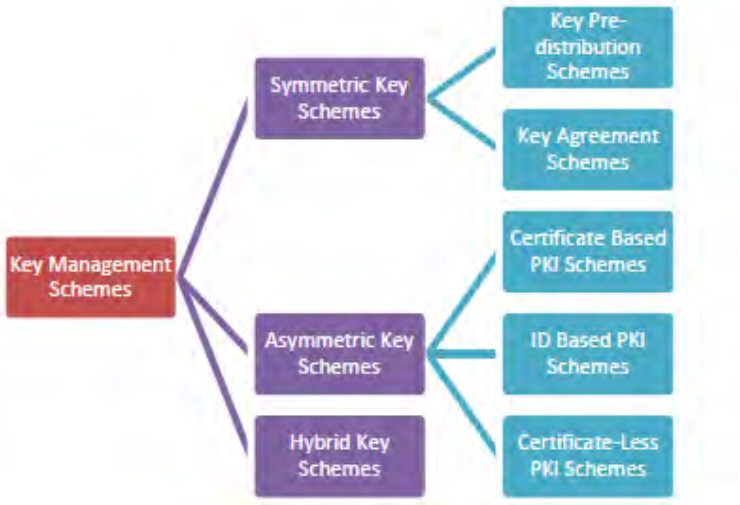



**Figure 2. : Classification des systèmes de gestion de clés**

La gestion de clés symétriques suit l'infrastructure de clé privée, qui établit des clés privées communes utilisées pour la cryptographie symétrique. Le schéma de clés symétriques est en outre catégorisé en schéma basé sur la pré-distribution de clés, dans lequel la clé secrète est délivrée par le tiers de confiance au nœud avant le déploiement et le second schéma est basé sur l'accord de clé, dans lequel les utilisateurs s'accordent mutuellement sur la clé privée commune après le déploiement.

Le schéma de gestion de clés asymétriques considère l'infrastructure de clé publique, qui fournit une paire de clés, c'est-à-dire une clé publique et une clé privée utilisées pour la cryptographie asymétrique. Le schéma de gestion des clés asymétriques est ensuite catégorisé en schéma basé sur le certificat pour lequel la troisième partie certifiée certifie l'authenticité de la clé publique des utilisateurs, en schéma basé sur l'identité pour lequel l'identité de l'utilisateur est utilisée comme clé publique et en schéma non basé sur le certificat, qui ne requiert aucun tiers de confiance, puisque l'utilisateur lui-même certifie sa clé publique.

Le schéma de gestion de clés hybride utilise à la fois des clés cryptographiques symétriques et asymétriques dans différentes phases du schéma.

## 2.4. La Révocation de Clés dans les MANETs et les WSNs

### 2.4.1 Définition de la Révocation de clés

La révocation de clé est généralement associée à l'invalidation de la clé d'un nœud en informant les autres nœuds du réseau qu'ils ne doivent plus communiquer avec le propriétaire de celle-ci [70].

Dans le contexte des MANETs et des WSNs, il existe diverses menaces dont la plus intéressante et la plus importante est probablement la subversion des nœuds [27]. Pour faire face à cette menace qui pourrait aboutir à la compromission de la confidentialité des clés, la compromission de l'authenticité des clés et l'usage non autorisé de clés, le processus de gestion des clés doit veiller à ce que les clés compromises soient annulées (révoquées).

### 2.4.2 Importance et Négligence de la Révocation

Il existe essentiellement deux approches pour sécuriser un MANET : l'approche proactive et l'approche réactive [71]. L'approche proactive tente en premier lieu de contrecarrer les menaces de sécurité, généralement au moyen de diverses techniques cryptographiques. D'un autre côté, l'approche réactive cherche à détecter les attaques à posteriori et à réagir en conséquence. En l'absence d'une ligne de défense claire, une solution de sécurité complète pour les MANETs devrait intégrer à la fois des approches proactives et réactives, et englober les trois composantes qui sont la prévention, la détection et la réaction.

Le composant de prévention dissuade l'attaquant en augmentant de manière significative la difficulté de pénétrer dans le système. Cependant, l'histoire de la sécurité a clairement montré qu'un système totalement exempt d'intrusion est infaisable, même si les mécanismes de prévention sont conçus avec soin. Ceci est particulièrement vrai dans les MANETs, constitués de dispositifs mobiles susceptibles d'être compromis ou capturés physiquement. Par conséquent, les composants de détection et de réaction qui détectent les intrusions occasionnelles et qui réagissent pour éviter les effets néfastes persistants sont indispensables pour que les solutions de sécurité fonctionnent en présence d'intrusions limitées.

Le composant de détection découvre les attaques en cours grâce à l'identification du comportement anormal présenté par des nœuds malveillants. Un tel comportement est détecté de bout en bout, ou par les nœuds voisins en écoutant le canal et en atteignant un consensus collaboratif. Une fois qu'un nœud attaquant est détecté, le composant de réaction effectue des ajustements dans les opérations de routage et d'acheminement, allant jusqu'à éviter le nœud dans la sélection de route pour l'exclure collectivement du réseau.

De cette réflexion, il découle que pour être complète, toute solution de gestion de clés proposée pour les MANETs ou les WSNs doit disposer d'un mécanisme qui se chargera d'exclure les nœuds compromis du réseau. Cette exclusion ou révocation de nœud compromis peut être réduite à celle de la révocation de sa ou de ses clés, car sans les clés cryptographiques valides, les nœuds malveillants ne

peuvent pas communiquer avec d'autres nœuds et participer au fonctionnement du réseau. De plus, tous les messages provenant de ces nœuds seront également ignorés par les pairs. Ainsi, en révoquant toutes les clés appartenant à un nœud compromis connu, on peut effectivement supprimer la présence de ce dernier du réseau.

Malgré son importance, la question liée à la révocation de clés dans les MANETs et les WSNs a malheureusement été prise en main tardivement. En effet pour les MANETS et particulièrement pour les WSNs, deux aspects importants de la gestion des clés sont la distribution des clés et la révocation des clés [18]. Dans ces réseaux, contrairement à la distribution des clés, qui a fait l'objet de nombreuses études [1-3], [5], [6], [8-10], [12-16], la révocation de clé a reçu relativement peu d'attention. De plus en plus, les chercheurs dans le domaine de la sécurité, et particulièrement dans le domaine de la gestion de la confiance et dans le domaine de la gestion des clés, se rendent compte de son importance et tentent de lui donner la place adéquate dans le processus de sécurisation de ces réseaux.

### 2.4.3 Contraintes de conception de solutions de révocation de clés

La révocation de clés dans les MANETs et les WSNs, partage les mêmes contraintes de conception avec la gestion des clés dans ces derniers. Il s'agit notamment de :

**L'absence d'infrastructure :** Avec les MANETs, puisque les solutions proposées sont généralement basées sur la cryptographie asymétrique, la révocation de clés dans ces réseaux est un type de mécanisme qui détermine si un certificat et la clé cryptographique correspondante d'un nœud du réseau doivent être révoqués. L'adoption des solutions de révocation de certificats traditionnelles proposées pour les réseaux filaires est associée avec des défis majeurs. En effet, avec ces dernières, pour obtenir une liste de révocation, chaque nœud, à la réception de chaque message signé, doit contacter individuellement le serveur central ou distribué de l'autorité de certification disponible sur le réseau câblé via la station de base ou le nœud *sink* pour récupérer la liste de révocation de certificat (*Certificate Revocation Lsit*, CRL). Alternativement, le serveur de CA peut envoyer la CRL et ses mises à jour suite à chaque révocation de clé directement vers tous les nœuds dans le MANET, conduisant à des envois par inondation et ainsi des charges de communication. Ce schéma n'a pas de protocole séparé, qui a été optimisé pour son implémentation sur un réseau ad hoc. Cependant après la révocation d'un certificat et mise à jour de la CRL, les utilisateurs doivent être avertis avec un système "push" ou "pull" [72].

Afin de mettre en œuvre un système "push", un serveur de CA pourrait diffuser périodiquement sa dernière liste CRL ou ses mises à jour à un ensemble d'abonnés. Le détenteur du certificat doit initier l'approche "pull" en vérifiant périodiquement auprès du serveur particulier d'où il a obtenu le certificat, ou avant de faire confiance à un certificat ou de le décrypter. Tout algorithme, qui nécessite une

révocation périodique, des mises à jour, que ce soit par un push ou par un accès pull, entraînera un délai entre le moment où le certificat est révoqué et le moment où les utilisateurs obtiennent des informations à ce sujet. Les schémas push sont gourmands en bande passante et sont voués à l'échec s'ils doivent diffuser une mise à jour de la CRL entière à chaque fois à un grand nombre d'abonnés. On peut essayer de réduire le besoin de révocation en assignant des certificats avec de courtes périodes d'expiration, mais cette approche conduit à un système avec beaucoup de frais généraux. Il appelle également à la distribution fréquente de certificats de remplacement [50] ;

**Les contraintes en ressources des nœuds :** Un autre défi majeur qui est plus pertinent pour les WSNs est celui de n'utiliser dans la conception du protocole que des ressources de calcul et de communication limitées ; C'est-à-dire que les protocoles de révocation doivent s'appuyer sur des primitives cryptographiques très simples et atteindre leur but avec un nombre limité de messages. Par exemple, les primitives efficaces incluent les fonctions de hachage et les arbres de hachage [99], l'évaluation de polynômes de faible degré etc. En revanche, les primitives gourmandes en énergie ou en mémoire, telles que les primitives basées sur des clés publiques ou les protocoles de consensus qui parviennent à des accords en présence d'adversaires malveillants et nécessitant des diffusions multiples sur l'ensemble du réseau, sont beaucoup moins souhaitables [18] ;

**La facilité de mise en œuvre des attaques :** Dans les MANETs, tout comme dans les WSNs, la révocation de clé pour les nœuds capturés pose de nombreux défis de conception puisque les protocoles sont effectués en présence d'adversaires actifs. Ces adversaires peuvent à la fois surveiller et modifier les messages réseau et, plus important encore, prétendre être des participants légitimes dans les protocoles eux-mêmes. Les nœuds capturés (et donc compromis) peuvent agir comme des substituts de l'adversaire dans un protocole de révocation, et peuvent coaliser pour en subvertir l'exécution (par exemple, ils pourraient bloquer le fonctionnement du protocole en épuisant les ressources des nœuds légitimes ou refuser d'exécuter les étapes clef du protocole). Ainsi, un défi spécifique dans la conception des protocoles de révocation est de réaliser la révocation des nœuds qui sont compromis par un adversaire malgré la participation active de cet adversaire dans le protocole.

## Conclusion

La gestion des clés est l'un des domaines les plus importants dans la sécurité des réseaux Ad hoc. Beaucoup de travaux ont été effectués dans le but d'avoir un schéma performant qui assure un niveau élevé de sécurité, optimise les métriques de performances et conserve l'énergie. Cependant majoritairement ces travaux se sont plutôt penchés vers la distribution de clés aux nœuds du réseau et l'établissement de clés entre ces derniers en négligeant un aspect fondamental qui est la révocation de clés en cas de compromission du nœud propriétaire.

De plus en plus les chercheurs se rendent compte de l'importance du domaine et tentent de lui donner la place qui lui convient. Ainsi, pour permettre une meilleure prise en main du problème que pose la révocation de clés dans les MANETs et les WSNs, nous nous proposons d'étudier et de comparer les différentes approches qui ont été proposées afin d'en faire ressortir les forces et les faiblesses et ouvrir des nouveaux axes de recherche.

# Chapitre 3 LA REVOCATION DE CLES DANS LES MANETS ET LES WSNS

La gestion des clés est un service cryptographique de base requis avant que tout autre service de sécurité ne puisse être déployé dans le réseau. En tant que composante de tout système de gestion des clés, la révocation de clé joue un rôle très important pour la sécurité et la robustesse de MANET et des WSNs.

Afin de faire connaitre les différentes approches qui ont été adoptées pour aborder le problème que pose la révocation de clés dans ces réseaux nous proposons, dans ce chapitre, une présentation des solutions qui ont été proposées.

## 3.1 Classification des Protocoles de Révocation de Clés dans les MANETs et les WSNs

Les protocoles de révocation de clés peuvent être classés de différentes façons [17], [20], [73], [74]. Cependant pour faire ressortir les spécificités et les limites de chaque type de réseau, nous nous inspirons de [17] et classons les protocoles de révocation proposés pour les MANETs et les WSNs en quatre classes. Ces classes basées sur l'implication des nœuds et des autorités centrales sont respectivement constituées des protocoles centralisés, distribués, décentralisés et hybrides :

- **Protocoles Centralisés :** dans ces protocoles, une autorité centrale est tenue de révoquer les nœuds compromis en supprimant les clés compromises et / ou en mettant à jour certaines clés. Par exemple, les clés de session partagées entre les nœuds ou les clés réseau utilisées dans la communication de groupe ;
- **Protocoles Distribués :** ces protocoles requièrent la collaboration de plusieurs nœuds dans le processus de révocation des clés compromises ;
- **Protocoles Décentralisés :** la décision de révoquer un nœud compromis revient à un et un seul nœud, sans l'implication d'une autorité centralisé ou le besoin de coopération entre les nœuds ;

- **Protocoles Hybrides :** ces protocoles sont basés sur le concept de groupage des nœuds et combinent et les méthodes centralisées et les méthodes distribuées pour accroitre leur efficacité et leur exactitude.

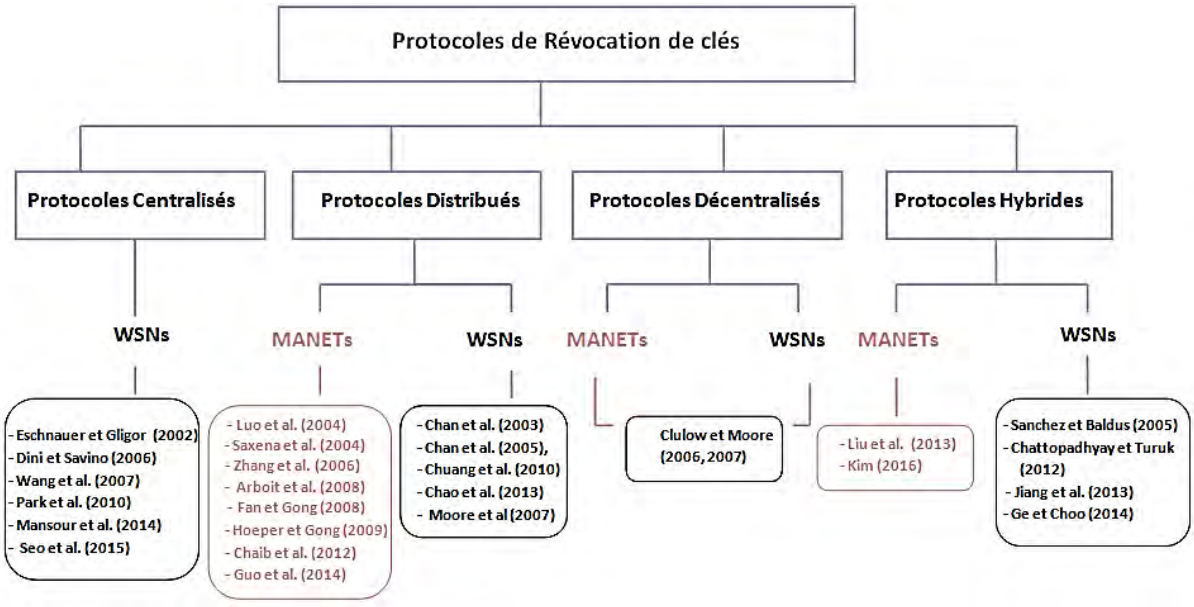



**Figure 3. : Classification des protocoles de révocation de clés dans les MANETs et les WSNs**

## 3.2 Les protocoles de Révocation de clés dans les MANETs

Pour tout mécanisme de gestion des clés dans les MANETs, il est souhaitable qu'il soit non centralisé, dynamique et efficace [34]. Afin de répondre à ces propriétés, les concepteurs proposent pour les MANETs des protocoles de révocation de clés basés, sur la cryptographie à clé publique pouvant appartenir à une des classes précitées, exceptée celle proposant des protocoles centralisés.

### 3.2.1 Les protocoles Distribués

Les principaux protocoles de révocation de certificats/clés distribués étudiés pour les MANETs sont les suivants : Luo et al. (2004) [6], Saxena et al. (2004) [8], Zhang et al. (2006) [10], Arboit et al. (2008) [77], Fan et Gong (2008) [79], Hoeper et Gong (2009) [28], Chaib et al. (2012) [82] et Guo et al. (2014) [83].

***Luo et al. (2004)***

Dans [6] et [75], Luo et al. se basent sur un schéma à seuil $(t, n)$ et proposent un mécanisme de gestion de clés localisé, qui distribue les capacités de la CA à tous les nœuds du réseau. Les auteurs décrivent aussi brièvement dans [6] un mécanisme de révocation de ticket localisé. Dans celui-ci, chaque nœud surveille son voisinage à un-saut et la révocation de ticket a lieu dans les deux cas ou

scénarios suivants : *i)* quand le nœud $i$ détecte le comportement malveillant d'un nœud voisin $j$ ou *ii)* quand le nœud $i$ collecte contre un nœud $j$ un nombre d'accusations supérieur à un seuil de révocation prédéfini. Dans le premier cas, après avoir marqué le nœud $j$ comme étant exclu, $i$ propage son accusation signée dans son voisinage à $m$-sauts. Dans le second cas, seules les accusations valides et lancées par les nœuds fiables (non révoqués) sont acceptées par $i$. $j$ sera révoqué une fois que le nombre d'accusations collecté contre lui atteint le seuil de révocation défini. Le nœud $j$ sera alors définitivement révoqué et exclu du réseau après une période $T_{ticket}$ car ne pourra plus bénéficier de service de mise-à-jour de ticket à l'issue de cette période.

***Arboit et al. (2008)***

Dans [76] Crépeau et Davis proposent le premier mécanisme de révocation de certificats auto-organisé. Leur protocole se base sur les accusations pondérées et fournit une protection contre les potentielles fausses accusations lancées par les nœuds malveillants. Plus tard, en 2008, Arboit et al. proposent dans [77] une version améliorée de leur première solution. Dans celle-ci, les auteurs utilisent l'idée de la théorie des jeux pour offrir une protection contre les fausses accusations en provenance de nœuds attaquants. Dans ce protocole, la fiabilité d'un nœud détermine le poids de ses accusations et de celles portées contre lui. Ainsi les accusations en provenance de nœuds dignes de confiance auront plus de poids que celles des nœuds moins dignes de confiance. En effet, lorsqu'un nœud est accusé par un nombre maximum de pairs, et a en même temps atteint le nombre maximum d'accusations autorisé, sa valeur de confiance ou sa fiabilité est alors à sa valeur minimale possible. Par conséquent le poids de ses accusations devient nul.

Avec leur mécanisme de révocation, toutes les accusations sont propagées par diffusion vers tout le réseau, et les chaines de hachage à sens unique sont utilisées pour garantir leur authenticité et leur intégrité. Le certificat d'un nœud est alors révoqué lorsque la somme des accusations pondérées portées contre celui-ci est supérieure ou égale à un seuil ou quotient de révocation configurable.

***Saxena et al. (2004) - ID-GAC***

Afin d'adresser certaines faiblesses relatives aux systèmes basés sur le certificat, Saxena et al. proposent dans [8] un mécanisme de révocation de membres basé sur l'identité. Leur protocole de révocation est similaire à celui décrit dans [75] et [6], car il utilise un système de partage à seuil $(t, n)$ pour distribuer la clé « *master key* » à tous les nœuds du réseau avant leur déploiement. Cependant à la différence du protocole décrit dans [6], ID-GAC propose un mécanisme de révocation qui, *i)*utilise une méthode de signature à seuil pour protéger les échanges pour la mise a jour de parts de clé, *ii)* propage les messages d'accusation à travers tout le réseau, *iii)* se base sur les listes de révocation de membres (*Member Revocation List* (MRL)) analogues aux listes de révocation de certificats.

***Zhang et al. (2006) - IKM***

Dans [10] Zhang et al. proposent un mécanisme de gestion de clés (IKM) utilisant la cryptographie basée sur l'identité. Dans celui-ci, les auteurs décrivent une nouvelle méthode de construction de paires de clés public/privée basée sur l'identité. Ils distribuent les fonctionnalités d'un TTP externe à $N$nœuds du réseau pour ainsi former un centre de génération de clés distribué (*Distributed Private Key Generators* (D-PKG)), implémenté à l'aide d'un système à seuil $(t, N)$. Dans leur protocole de révocation, chaque nœud observant le comportement malveillant d'un autre nœud envoie contre ce dernier un message d'accusation aux D-PKGs pré-assignés. Celui-ci est signé et encrypté a l'aide d'algorithmes basés sur l'identité. Lorsque le nombre d'accusations portées contre un nœud atteint le seuil de révocation dans un intervalle de temps prédéterminé, les D-PKGs collaborent pour révoquer ce nœud attaquant par le biais d'un système de signature à seuil. Pour ce faire, le D-PKG leader calcule à partir de $t$ valeurs de révocation partielles reçues, la valeur complète de révocation qui sera par la suite envoyée par boradcast à tous les nœuds du réseau.

***Hoeper et Gong (2009)***

Dans [78], Hoeper et Gong proposent pour les systèmes basés sur l'identité un mécanisme de révocation de clés auto-organisé. Les auteurs présentent un nouveau format de clé publique basé sur l'identité incluant l'identité du nœud, la date d'expiation de la clé et son numéro de version. Ceci dans le but de pouvoir délivrer de nouvelles clés pour la même identité après révocation. Dans le but de rendre plus flexible leur solution et de renforcer sa robustesse, les auteurs proposent plus tard en 2009 une version améliorée de leur protocole de révocation de clés [28]. Dans celle-ci, chaque nœud surveille son voisinage à un-saut. En cas de détection du comportement malveillant d'un nœud voisin $j$, le nœud $i$ propage vers son voisinage à $m$-sauts un message d'accusation porté contre $j$. Les messages d'accusation envoyés sont sécurisés par le biais de clés secrètes pré-partagées, obtenues à partir d'un protocole d'accord de clés non interactif basé sur l'identité.

Leur mécanisme de révocation donne en plus à chaque nœud la possibilité de révoquer sa propre clé privée en cas de compromission de celle-ci en propageant vers tout le réseau un « *Harakiri* » message. Ainsi la clé publique d'un nœud $j$ est considérée par le nœud $i$ comme étant révoquée si au moins une des conditions suivantes est vérifiée :

*i)* le nœud $i$ observe le comportement malveillant de $j$ à travers le « *Neighborhood Watch* » ;

*ii)* le nœud $i$ reçoit un message « *Harakiri* » en provenance de $j$ ;

*iii)* le nœud $i$ reçoit au moins $\delta$ accusations portées contre $j$ par des nœuds fiables (c'est à dire non révoqués) de son voisinage à $m$-sauts, $\delta$ étant un seuil de révocation paramétrable. Afin de combattre les potentielles fausses accusations lancées par les voisins à $l$-sauts ($2 \ll l \ll m$) coalisants, les auteurs proposent d'adopter le vote majoritaire basé sur le paramètre ε.

***Fan et Gong (2008)***

Dans [79] Fan et Gong proposent une solution de révocation de clés comportant cinq phases notamment : initialisation du réseau, surveillance du voisinage à un saut, propagation sécurisée de l'information, filtrage de fausses accusations et fourniture de réponses à propos du comportement d'un nœud. Leur proposition est similaire à celle décrite dans [28], car elle est auto-organisée, est sécurisée par le biais de la cryptographie basée sur l'identité, s'appuie sur la surveillance du voisinage à un saut et propage les messages d'accusation vers le voisinage à $m$-sauts. Cependant les auteurs de [79] ont attiré l'attention sur le fait que certains nœuds peuvent accidentellement avoir un comportement malveillant dû, par exemple à une incapacité temporaire de transférer les paquets. Ainsi ils jugent qu'il est plus raisonnable de continuer à collecter des informations sur le comportement de ce nœud plutôt, que de le considérer immédiatement comme étant malveillant et de l'exclure du réseau en révoquant sa clé.

Inspirés par ces observations, Fan et Gong proposent un protocole de révocation de clés auto-organisé, basé sur la combinaison du modèle multinomial de *Dirichlet* [80] et sur la cryptographie basée sur l'identité. Dans ce protocole, les nœuds sont classés en trois catégories selon leur comportement à savoir : nœuds au bon comportement, nœuds au comportement suspect et nœuds au comportement malveillant. Le modèle de *Dirichlet* est utilisé pour quantifier l'incertitude sur le comportement des nœuds explicitement avec une probabilité. Chaque nœud met à jour ses connaissances sur le statut de la clé d'un autre nœud pair en observant son voisinage à un saut et en analysant les rapports, relatifs aux observations des autres nœuds. Deux lignes de défense sont alors dressées pour contrer d'éventuelles fausses accusations lancées par les nœuds malveillants. Un test de déviation, basé sur le modèle statistique des rapports, est d'abord utilisé pour filtrer les fausses accusations. Si le rapport du nœud accusateur passe ce test de déviation, le nœud récepteur du rapport applique ensuite la théorie de la croyance de *Dempster-Shafer* [81] pour mettre à jour ses connaissances à propos du comportement du sujet en question. Le nœud récepteur du rapport donnera une réponse à partir de l'analyse de l'information collectée. Le lecteur est invité à se référer à l'Algorithme 1 pour une description plus détaillée du protocole de révocation de clés.

Algorithme 1 : Révocation de clés auto-organisée pour les MANETs

**Etape 1.** Initialisation du réseau

- *Génération des system paramètres système*
- *Enregistrement des nœuds du réseau*
- *Classification du comportement des nœuds*

**Etape 2.** Surveillance du voisinage

- *Surveillance du voisinage et génération de la matrice d'observation*
- *Mise à jour de statut de nœuds avec les observations directes*

**Etape 3.** Dissémination d'informations authentifiées

- *Propager les observations directes des nœuds vers le voisinage à $m$-sauts de façon authentifiée en utilisant une fonction de hachage*

**Etape 4.** Filtrer les fausses accusations

- *Filtrer statistiquement les potentielles fausses déclarations*
- *Mise à jour de statut de nœuds en se basant sur la théorie de Dempster-Shafer*

**Etape 5.** Réponse à plusieurs niveaux pour les nœuds malveillants

- *Révoquer les clés des nœuds montrant un comportement malveillant*
- *Cesser la communication avec les nœuds présentant un comportement suspect et continuer à observer leur comportement pour une décision ultérieure*

***Chaib et al. (2012)***

Dans [82] Chaib et al. mettent en exergue la vulnérabilité des protocoles de révocation distribués basés sur le vote avec seuil face aux attaques par coalition de nœuds. Ils attirent l'attention sur le fait que ces protocoles ne peuvent résister à une attaque par révocation [18] en cas de présence d'un nombre de nœuds attaquants, coalisants dont le nombre dépasse le seuil de révocation préétablit. Pour répondre à ce problème, Chaib et al. proposent une nouvelle approche de révocation distribuée, qui se défait de toute utilisation d'un seuil de révocation. Pour ce faire, les auteurs combinent les avantages des propositions antérieures pour présenter une approche hybride présentant les caractéristiques suivantes :

*i)* un seul message d'accusation peut conduire à la révocation du nœud accusé sans punition du nœud accusateur ;

*ii)* le protocole empêche la révocation d'un nombre élevé de nœuds légitimes par la seule présence dans le système d'un nombre limité de nœuds coalisant.

Chaib et al. supposent que chaque nœud est doté d'un système de détection d'intrusion et maintient une liste d'accusations et une liste de révocation, aussi appelé *Blacklist*. Après détection d'un

comportement malveillant, chaque nœud notifie tous les nœuds de son voisinage en envoyant par diffusion un message d'accusation signé. Afin de contrer les fausses accusations, les auteurs suggèrent l'évaluation de chaque accusation reçue par le biais d'une fonction $d(x)$ appelée *dishonesty function.* La valeur renvoyée par cette fonction dépend du nombre d'accusations portées contre le nœud $x$, et du nombre d'accusations lancées par le nœud $x$. Cette valeur permettra de décider s'il faut accepter une accusation et révoquer le nœud en question ou non. Plus explicitement, après réception d'une accusation, $d(a)$ est évaluée pour le nœud accusateur et $d(e)$ est évaluée pour le nœud accusé ; L'acceptation de cette accusation et la révocation du nœud accusé, c'est-à-dire son placement dans la *Blacklist*, ne se fera que lorsque $d(a) \leq d(e)$.

***Guo et al. (2014)* [83]**

Dans [83], Guo et al. proposent un mécanisme de révocation de certificat similaire à celui décrit dans [76] et [77] car :

*i)* Il est auto-organisé ;
*ii)* Il se base sur les accusations pondérées ;
*iii)* Chaque nœud maintient deux tables (une table d'accusations et une table de profils) stockant des informations qui seront utilisées pour déterminer si un nœud doit être révoqué ou non.

Pour limiter les fausses accusations, leur protocole préconise :

*i)* L'acceptation d'une accusation uniquement si le poids du nœud accusateur est supérieur ou égal au poids du nœud accusé ;
*ii)* L'augmentation et la diminution respective de la valeur de confiance d'un nœud accusateur et du nœud accusé d'une certaine valeur ;
*iii)* Le retrait de la fonction d'accusation d'un nœud dès la réception de la première accusation portée contre lui.

Dans ce protocole de révocation, les nœuds observent leur voisinage à un saut. Après détection du comportement malveillant d'un voisin $j$, le nœud $i$ diffuse par broadcast à travers tout le réseau un message d'accusation. Après réception et vérification de ce message, le nœud récepteur met à jour sa table d'accusations et sa table de profils. Le certificat du nœud $j$ sera revoqué si au bout de la période $T$ la valeur cumulative des accusations pondérées reçues contre lui est supérieure ou égale à un seuil de révocation configurable. Dans le but de limiter les dommages causés par les nœuds attaquants, le protocole préconise en plus, suite à la révocation d'un nœud :

*i)* La compensation des nœuds qui avaient été accusés par ce dernier ou qui ont porté une accusation contre ce dernier ;

*ii)* La restauration de la fonction d'accusation d'un nœud après révocation de tous les nœuds qui avaient porté une accusation contre lui.

La figure 3.2 illustre l'architecture de Guo et al. (2014) [83] en classant les nœuds en trois catégories selon leur fonction : nœuds normaux (*Normal nodes*), nœuds blancs (*White nodes*) et les nœuds noirs (*Black nodes*). *L, M, B, N, W* représentent des ensembles comportant respectivement les nœuds légitimes, malveillants, noirs, normaux et blancs

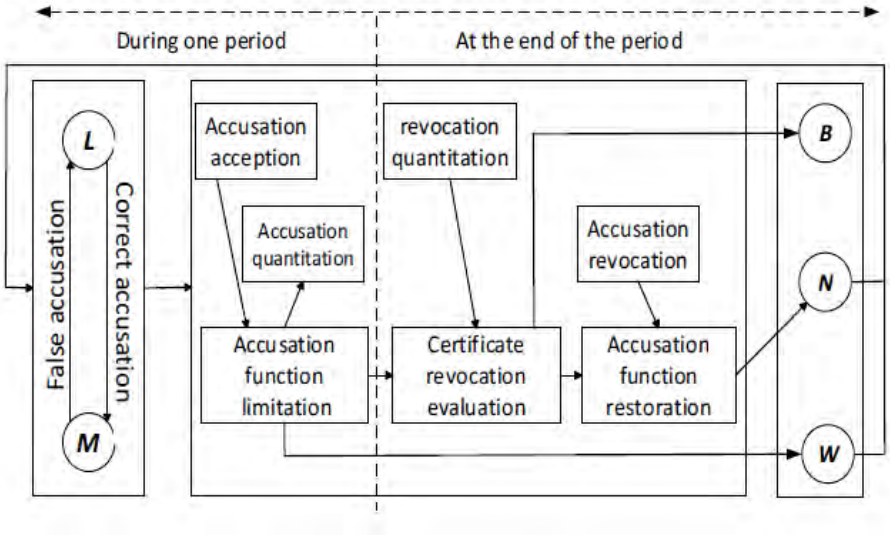



**Figure 3. : Architecture de Guo et al. (2014) [83]**

### 3.2.2 Les protocoles Décentralisés

Nous avons pour cette catégorie le protocole Clulow et Moore (2006, 2007). Nous notons que cette solution a été conçue originalement pour les systèmes auto-organisés, mais la version étendue proposée par la suite en 2007 vise les réseaux ad hoc en général.

***Clulow et Moore (2006, 2007)***

En 2006, Clulow et Moore ont proposé dans [84] une stratégie de révocation entièrement décentralisée qu'ils qualifient de « suicide pour le bien commun ». Avec ce protocole, la décision de révocation est prise par un seul nœud. Si le nœud *i* est compromis et détecté par un autre nœud *j*, celui-ci diffuse une note de suicide comprenant à la fois son identité et celle du nœud ciblé. Il existe deux façons de signer la note, en utilisant la cryptographie à clé publique ou la cryptographie symétrique.

Dans l'approche basée sur la cryptographie à clé publique, la clé publique de $j$ est utilisée pour vérifier sa signature. Dans celle utilisant la cryptographie à clé symétrique, avant le déploiement, une clé unique de révocation est générée pour chaque nœud et un arbre de hachage, qui utilise la valeur de hachage de l'identifiant du nœud et la clé correspondante en tant que feuille. Ensuite, la clé de révocation, la valeur d'authentification racine et les $\log n$ valeurs d'authentification de chemin ($n$ étant le nombre de nœuds dans le réseau) sont distribuées à chaque nœud. Le nœud $j$ diffuse la note signée par sa clé de révocation en premier, avant de révéler la clé et les valeurs d'authentification de chemin. Après avoir reçu la note, les autres nœuds la vérifient et ajoutent $j$ et $i$ dans une liste noire (blacklist) si la vérification réussit.

Dans l'approche de la clé symétrique, la valeur de hachage de la racine est calculée à l'aide de la clé de révocation et des valeurs d'authentification de chemin, et la valeur de hachage calculée est comparée à la valeur racine stockée en mémoire.

Le mécanisme est conçu avec un délai dans le but d'éviter une seule note de suicide faite par plusieurs nœuds. Plus spécifiquement, si les nœuds se trouvent dans la même zone, ils diffusent des offres horodatées. Après un certain délai, le nœud avec la première offre diffuse la note de révocation. Si les nœuds se situent dans des zones différentes, on suppose qu'ils utilisent des techniques pour empêcher les attaques sybil et les attaques par réplication de nœuds.

Un an plus tard, Moore et al. décrivent officiellement dans [27] la stratégie de suicide (Clulow et Moore, 2007) ainsi qu'une analyse détaillée de sécurité et de performance. Ils présentent alors trois façons de mettre en œuvre la stratégie de suicide.

Le moyen le plus simple selon eux consiste à impliquer une autorité centrale. Lorsqu'un nœud compromis $i$ est détecté, le nœud $j$ envoie une note de suicide, qui inclut les deux identités, à l'autorité centrale. Pour signer la note, $j$ utilise la clé secrète, partagée avec l'autorité centrale. Après réception de la note, cette dernière la vérifie et informe les autres nœuds du réseau.

S'il n'y a pas une autorité centrale fiable, des mécanismes de suicide distribués sont introduits. Après détection d'un nœud compromis $i$, le nœud $j$ diffuse une note de suicide signée comprenant les deux identités. Après réception de cette note, les autres nœuds la vérifient, et en cas de succès les liens avec $j$ et $i$ seront coupés, et ces derniers seront placés sur une liste noire. Pour signer la note, il existe deux options. La première option consiste à utiliser la cryptographie à clé publique. Pour chaque nœud, une paire de clés publique/privée est générée, puis la clé publique est signée avec sa clé privée puis les envoie avec sa clé publique. Le message de diffusion comprend la note de suicide ainsi que le certificat de la clé publique. Lors de la réception de la note, les autres nœuds vérifient la clé publique avec la clé publique du propriétaire pré-chargée dans leur mémoire. Dans la deuxième méthode chaque

nœud est pré-chargé avec une clé de signature privée et une clé publique correspondante. Un arbre de hachage avec la clé publique de chaque nœud, en tant que feuille, est généré et la valeur de la racine, ainsi que les $\log n$ valeurs d'authentification de chemin d'accès, sont alors envoyées à chaque nœud. Le nœud $j$ diffuse la note de suicide signée avec les valeurs d'authentification du chemin d'accès.

Pour éviter de multiples demandes de suicide pour un nœud compromis, le délai de révocation est utilisé. Avec une approche centralisée, l'autorité centrale attend des doublons après avoir reçu la première note de suicide, puis diffuse l'une d'entre elles vers les autres nœuds du réseau. Par contre avec une approche décentralisée, un nœud attend une période aléatoire avant d'envoyer sa note. Le nœud dépose sa note lorsqu'il reçoit une autre note pendant la période. Sinon, il diffuse sa note lorsque le délai expire. Cependant, comme ce système n'empêche pas l'émission de plusieurs notes, la méthode de combinaison avec l'horodatage est présentée. Après avoir attendu un moment précis, le nœud A diffuse la note de suicide signée avec un horodatage. Tous les nœuds attendront alors une période de temps préalablement définie. Lorsque la minuterie expire, ils vérifient la note avec l'horodatage le plus ancien.

Les solutions présentées dans [27] sont étendues pour résoudre le problème des mécanismes de détection avec une incertitude élevée et une faible précision. Une solution à ce problème consiste à attribuer une probabilité au nœud malveillant. Supposons qu'un nœud détecte un comportement probablement malveillant il peut y attribuer une probabilité $p$ (par exemple, mauvais avec $p = 0{,}7$). Lorsque la valeur assignée dépasse un seuil, une action de suicide est effectuée. Une meilleure solution est d'ajouter la probabilité liée à cette incertitude à la note de suicide. Ainsi, la décision de suicide peut être prise par l'un des nœuds participants en fonction de la probabilité incluse dans la note.

### 3.2.3 Les protocoles Hybrides

Nous avons principalement Liu et al. (2013) [73], et Kim (2016) [87].

***Liu et al. (2013)***

Pour améliorer leurs propositions antérieures [85] et [86], Liu et al. proposent dans [73] un mécanisme de révocation de certificat nommé CCRVC (*Cluster-based Certificate Révocation with Vindication Capability*), qui se base sur le concept de cluster. Leur protocole se base sur une tierce partie de confiance, ou CA, qui a pour rôle de délivrer des certificats aux nœuds avant leur déploiement. La CA représente aussi l'entité qui se charge de la mise à jour des listes WL (*Warning List*) et BL (*BlackList*) utilisées pour stocker respectivement les informations relatives aux nœuds accusateurs et aux nœuds accusés avant de les propager dans tout le réseau.

Dans leur solution les nœuds sont classés en trois catégories, selon leur fiabilité : nœud normal (*normal node* ), nœud averti (*warned node*), nœud révoqué ( *revoked node*). Les nœuds normaux sont ceux qui n'ont encore lancé aucune accusation contre leurs pairs et sont considérés comme étant très fiables et ont la possibilité de lancer des accusations et de s'auto déclarer CH ou membre de cluster. Les nœuds avertis listés dans la WL présentent une faible fiabilité car sont considérés comme étant suspects. Ils ont la possibilité de communiquer avec leurs voisins mais avec une certaine restriction. Par exemple le droit d'accuser leurs nœuds voisins leur est retiré afin d'éviter d'autres abus. Les nœuds révoqués qui sont listés dans la BL sont considérés comme étant des attaquants et sont alors révoqués et exclus du réseau.

Dans leur mécanisme de révocation, tout nœud *j* détectant la présence d'un attaquant *i* dans son voisinage vérifie en premier la BL locale, afin de voir s'il s'agit d'un nœud déjà détecté ou non. Si ce nœud *i* n'y est pas, le nœud *j* envoie à la CA un message d'accusation *(Accusation Packet (AP)).* Après réception du premier AP, la CA vérifie la légitimité du nœud accusateur par le biais de son certificat. En cas de succès, le nœud *i* est considéré comme étant malveillant et est alors ajouté à la BL. Cependant, le nœud accusateur est par la même occasion ajouté à la WL. La procédure de révocation se terminera une fois que la CA aura propagé à travers le réseau un message de révocation incluant les WL et BL.

Les auteurs considèrent que chaque nœud CH peut détecter toutes les attaques de nœuds membres de son cluster. Par conséquent le CH a la possibilité de restaurer tout nœud considéré comme étant faussement accusé. Pour ce faire, le CH envoie à la CA un message de restauration ou *(Recovery packet (RP)).* Cette dernière retirera le nœud en question de la BL, après vérification de la légitimité du nœud CH, avant de propager cette dernière information à travers tout le réseau. Il faut noter qu'avec ce procédé, l'accroissement du nombre de nœuds malveillants dans le réseau s'accompagne de l'accroissement du nombre de nœuds normaux listés dans la WL. En conséquence il n'y aura plus assez de nœuds normaux pour lancer des accusations contre les nœuds attaquants. Ce scénario affectera alors la fiabilité du protocole. Pour répondre à ce problème, les auteurs proposent en outre un mécanisme de récupération *(node release)*. Dans celui-ci, la CA continue de recevoir les accusations pendant une certaine période de temps et collecte pour chaque nœud accusé le nombre d'accusations reçues. Pour chaque nœud ce nombre sera alors comparé à un seuil à la fin de cette période. Ainsi un nœud accusé sera considéré comme étant un véritable attaquant si, à l'issue de cette période, le nombre d'accusations collectées contre lui est supérieur ou égal au seuil. Dans ce cas le nœud accusateur listé dans la WL en sera retiré pour être de nouveau considéré comme un nœud légitime et fiable avec toutes ses fonctions.

***Kim (2016)***

Dans [87] Kim propose une solution de révocation de certificat, basée sur la théorie des jeux et sur le concept de cluster. L'auteur propose un nouveau modèle de décision basé sur le vote pondéré, et propose un mécanisme de restauration de certificat pour apporter une protection contre les fausses accusations. Le modèle adopté présente en premier lieu une méthode pour la quantification du poids des nœuds et la détermination du quota de votes. Le poids de chaque nœud est évalué à travers sa fiabilité et l'historique de son comportement. Le quota de révocation est estimé séparément pour chaque cluster et de façon dynamique et en se basant sur la technique itérative d'apprentissage décrite dans [88], [89]. Ainsi pour révoquer un nœud malveillant membre d'un cluster, en se basant sur le modèle de jeux pondéré, le nœud CH (*cluster Head node*) diffuse un message d'accusation vers tous les membres de son cluster. Chaque nœud membre répondra alors à ce message en envoyant au CH son vote. Les nœuds votant pour la révocation du nœud $i$ adopteront alors la stratégie PV (pro-vote) et les nœuds votant contre la révocation du nœud $i$ adopteront la stratégie CV (con-vote). Le CH collectera les votes avant de les transmettre à la CA. Le poids de chaque nœud ayant voté pour la révocation de $i$ sera alors réajusté. Si le quotient de votes PV reçus représentant la somme des poids des nœuds votant pour la révocation du nœud $i$ est supérieur au quota de révocation, le CH révoquera ce dernier en envoyant par diffusion un message de révocation à tous les membres de son cluster. Pour lutter contre les fausses accusations, l'auteur suggère de procéder de la même manière pour la restauration du certificat d'un nœud, jugeant qu'il est victime de fausses accusations.

## 3.3 Les protocoles de Révocation de clés dans les WSNs

En raison des contraintes plus aigües en termes de ressources des nœuds, la presque totalité des solutions proposées pour les WSNs sont basées sur la cryptographie symétrique.

### 3.3.1 Protocoles Centralisés

Pour cette catégorie, nous avons les protocoles Eschnauer et Gligor (2002) [91], Dini et Savino (2006) [92], Wang et al. (2007) [93], Park et al. (2010) [95], Mansour et al. (2014) [96], Seo et al. (2015) [29]**.**

***Eschnauer et Gligor (2002)***

La question liée à la révocation de clés dans les réseaux de capteurs a d'abord été abordée en 2002 dans [91] par Eschenauer et Gligor qui ont présenté le premier protocole de révocation de clé centralisé pour les WSNs. Les auteurs supposent qu'un nœud contrôleur mobile ayant une grande portée de communication (par exemple la station de base (SB)) effectue le processus de révocation.
Avant de procéder à la révocation, ce contrôleur génère une clé de signature, la crypte à l'aide d'une clé de paire (pré-assignée à chaque nœud) et l'envoie par unicast à chaque nœud non compromis. De cette façon, pour révoquer un nœud, le contrôleur diffusera par broadcast un message signé, listant tous les

identifiants de nœuds compromis. Après réception et vérification de ce message, chaque nœud supprime les clés correspondantes de son trousseau de clés.

***Dini et Savino (2006)***

Dini et Savino proposent dans [92] un autre protocole de révocation clé centralisé pour les WSNs, qui suit le paradigme de communication de groupe. Ce protocole utilise un *Key service* (KS), qui est responsable de la procédure de révocation. Lorsqu'un nœud est suspecté d'être compromis, le KS l'oblige à quitter le groupe en révoquant la clé de groupe en cours, et en redistribuant une nouvelle clé à tous les autres nœuds du groupe. Pour accroître l'efficacité et la scalabilité de leur proposition, les auteurs ont mis au point un mécanisme basé sur la hiérarchie des clés logiques (*Logical Key Hierarchy, LKH*). À cet effet, le KS maintient un arbre de révocation, utilisé pour supprimer le nœud compromis (voir figure 3.3).

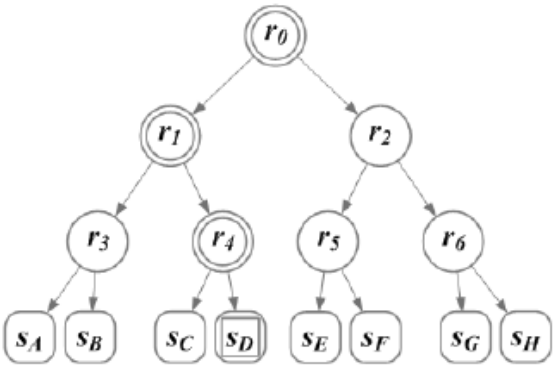



**Figure 3. : Arbre de révocation**

Chaque nœud détient une clé privée $k_j^p$ partagée avec le KS. En plus de cette clé privée, chaque nœud $j$ stocke un trousseau ou un sous-ensemble $keyRing(j)$ de clés courantes associées aux nœuds internes situés sur le chemin d'accès de la racine à la feuille associée à $j$. Si $Path\ (j)$ est l'ensemble de ces nœuds internes $r_n$, ce trousseau ou un sous-ensemble de clés se note [92] :

$$keyRing(j) = \left\{k_{cur}^{r_n} \middle| r_n \in Path(j)\right\} \qquad (3.1)$$

Une fois qu'un nœud compromis $i$ est detecté, toutes les clés appartenant à son trousseau doivent être renouvelées. Pour ce faire, le KS diffuse un message incluant la prochaine clé $k_{nxt}^{r_n}$ de $r_n$, $r_n \in Path(i)$ selon la procédure suivante :

- Si le nœud enfant de $r_n$ est une feuille qui n'est pas associée à $k_i^p$ alors, $k_{nxt}^{r_n}$ est chiffré avec la clé privée du membre correspondant du groupe ;
- Si le nœud $r_c$ enfant de $r_n$ est un nœud interne et n'est pas dans $Path(i)$ alors, $k_{nxt}^{r_n}$ est chiffré avec $k_{cur}^{r_c}$ ;
- Si le nœud $r_c$ enfant de $r_n$ est un nœud interne inclus dans $Path(i)$ alors, $k_{nxt}^{r_n}$ est chiffré avec $k_{nxt}^{r_c}$.

Après la fin de cette procédure, la clé actuelle $k_{cur}^{r_n}$ de chaque nœud interne appartenant à $Path(i)$ est remplacée par $k_{nxt}^{r_n}$. De cette façon, l'ensemble des clés actuelles associées aux nœuds internes de l'arbre de révocation est mis à jour, et la feuille associée à $k_i^p$ est retirée de l'arbre de révocation.

Dans ce mécanisme, chaque nœud maintient une chaîne de clés à des fins d'authentification. Ainsi, lors de la réception et du décryptage d'un message de renouvellement de clé, un membre du groupe peut vérifier son authenticité en déterminant $H(k_{nxt}^{r_n})$, $H$ étant une fonction de hachage à sens unique. Si $k_{cur}^{r_n} = H(k_{nxt}^{r_n})$ alors, le message est authentifié.

***Wang et al. (2007) – KeyRev***

Dans [93], Wang et al. proposent un autre protocole de révocation clé centralisée nommé *KeyRev*, qui utilise les techniques de mise à jour de clés afin de désactiver les clés appartenant au nœud compromis. Leur protocole ne dépend pas d'un système de distribution de clé spécifique, mais les auteurs supposent que le mécanisme de distribution de clés aléatoires proposé par Eschenauer et Gligor dans [91] est utilisé afin de décrire leur proposition. Dans *KeyRev*, les auteurs proposent d'utiliser uniquement deux types de clés à savoir : une clé de cryptage $k_{encr}$ et une clé MAC (*Message Authentication Code*) $k_{mac}$ pour sécuriser les communications dans le réseau. Ces clés sont générées de sorte que chacune d'elles soit liée à une clé de session $k_j$. Ces deux clés seront changées quand la clé de session $k_j$ sera mise à jour. Pour ce faire, Wang et al. suggèrent d'utiliser un système de distribution de clé de session basé sur [94].
Dans celui-ci, plus précisément dans sa phase de configuration, un *setup server* se charge pour chaque session, de fournir à chaque nœud *j* le secret dérivé à partir d'un polynôme de masquage $h_j(x)$ de degré $2t$ choisi aléatoirement par le *setup server*. Ce serveur fournit également ce polynôme à la SB. Dans la phase de diffusion de ce protocole, la SB diffuse aux nœuds non révoqués un message dont le format est : $B = \{R\} \cup \{P_j(x)\} \cup \{Q_j(x)\}$, où $P_j(x)$ et $Q_j(x)$ sont obtenus en utilisant la liste de nœuds révoqués durant la session *j* formant l'ensemble reçu $\{R\}$, et un polynôme $p_j(x)$ de degré $t$ choisi aléatoirement lors de la phase de configuration par le *setup server*. Pour obtenir la nouvelle clé de

session, tout nœud non révoqué $i$, recevant un tel message de diffusion, calculera les polynômes $P_j(x)$ et $Q_j(x)$ au point $i$. Par conséquent, il est possible d'empêcher les nœuds compromis de dériver $k_{encr}$ et $k_{mac}$ en les empêchant d'obtenir la nouvelle clé de session. Ainsi, les nœuds compromis peuvent être retirés du réseau en toute sécurité.

***Park et al. (2010)***

Pour renforcer la sécurité de *KeyRev*, Park et al. introduisent l'idée de sessions dynamiques dans [95] afin de réduire la durée de vie des nœuds compromis dans le réseau. Les auteurs proposent de gérer l'intervalle de temps existant pour les clés de session, qui comprend deux parties : le temps de révocation et le temps actif.

Durant le temps de révocation, la SB génère des clés de session, avant de les distribuer à chaque nœud du réseau en utilisant la technique de mise à jour de clé de session, proposée dans [93] par Wang et al..

En ce qui concerne le temps actif, l'intervalle est ensuite divisé en plusieurs niveaux de session (cinq niveaux dans [95]). Combiné avec les concepts de la taille dynamique de la session et du système événementiel, la SB peut décider de la prochaine taille de la session, et distribuer la nouvelle clé de session immédiatement lorsqu'un nœud compromis est détecté.

***Mansour et al. (2014)***

En 2014, Mansour et al. proposent dans [96] des protocoles de renouvellement et de révocation de clés pour les WSNs, basés sur le cryptage symétrique et la cryptographie de la courbe elliptique (ECC).

Avant le déploiement, chaque nœud $i$ est pré-chargé avec la clé publique de la SB, notée $pk(S)$ et sa paire de clés publique/privée, désignées respectivement par $pk(i)$ et par $sk(i)$. Dans ce mecanisme, $pk(i) = sk(i) * G$ où $G$ est un générateur de points de la courbe elliptique. Chaque nœud $i$ peut calculer une clé partagée avec la SB en utilisant une variation de l'échange de clés *Diffie-Hellman* sans interaction. Tous les nœuds du réseau partagent une clé de réseau $NK$.

Dans le protocole de révocation les auteurs supposent que la SB peut identifier les nœuds compromis à partir du système de détection d'intrusion. Si cette dernière détecte les voisins compromis du nœud $i$, elle envoie un message de révocation à $i$. Le message comprend une liste de nœuds malveillants et une information $n_s$ cryptée à l'aide de la clé $k_{DH}(S, i)$. Après réception du message, $i$ supprime les clés de session partagées avec tous les nœuds voisins listés dans le message, et renvoie $n_s$ chiffré par la clé partagée avec la SB pour confirmer la réception du message de

révocation. La SB doit également renouveler la clé de réseau lorsque des nœuds sont compromis. La SB envoie alors à chaque nœud non compromis $j$, un message contenant une nouvelle clé de réseau $NK'$ et une information $n_{(s,j)}$ et chiffrée par $k_{DH}(S,j)$. Après avoir reçu le message, le nœud $j$ renvoie $n_{(s,j)}$ chiffrée par la clé partagée avec la SB.

***Seo et al. (2015)***

Dans [29], Seo et al. présentent pour les WSNs dynamiques une solution de gestion des clés efficace nommée (*CL-EKM : CertificateLess-Effective Key Management*). Celle-ci se base sur la cryptographie à clé publique, se défait de toute utilisation de certificats et s'appuie sur le concept de cluster (voir figure 3.4 pour la structuration du réseau).

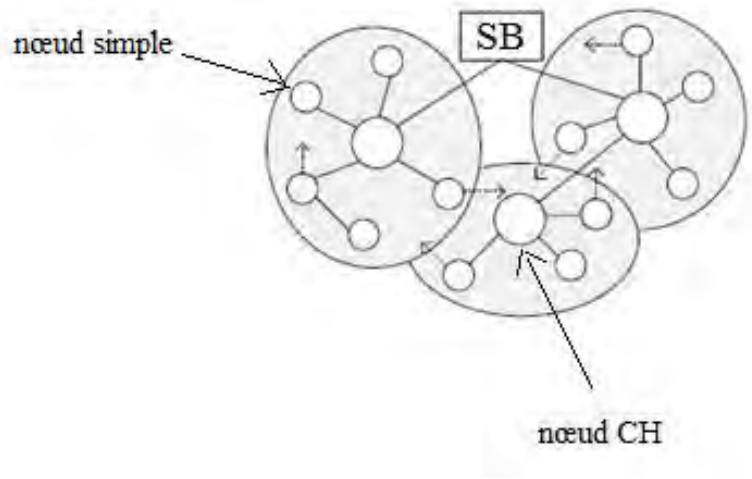



**Figure 3. : Structuration du réseau en cluster**

Dans leur solution de gestion de clés, la clé privée complète de l'utilisateur est une combinaison d'une clé privée partielle générée par un centre de génération de clé (KGC) et de la valeur secrète de l'utilisateur. L'organisation spéciale de la paire de clés publique/privée complète supprime le besoin de certificats. En effet, pour fournir dynamiquement l'authentification des nœuds et permettre l'établissement d'une clé de paire entre ces derniers, les auteurs s'appuient sur le protocole CL-HSC (*Certificateless Hybrid SignCryption scheme*) qu'ils avaient proposé antérieurement dans [97]. Dans CL-EKM, en raison des propriétés de CL-HSC, la clé de paire peut être efficacement partagée entre deux nœuds sans nécessiter des opérations couteuses de couplage (*pairing*) et l'échange de certificats.

La procédure décrite dans CL-EKM fournit un processus de révocation de clés, qui traite aussi bien le cas de la compromission d'un nœud simple que celui de la compromission d'un nœud CH.

Lorsqu'un nœud simple $i$ est compromis, la SB génère un message de révocation $E_{k_j^0}(compNode, i)$ chiffré, comportant l'identité de $i$ et l'envoie à chaque nœuds CH $j$ dans le réseau. Après réception et décryptage du message par le biais de la clé personnelle $k_j^0$ partagée entre la SB et le nœud $j$, le nœud CH $j$ met à jour la liste de révocation qui se rapporte à son cluster et ignore les clés associées à $i$ si elles existent. Pour compléter la révocation de clés du nœud $i$, chaque nœud CH $j$ exécute en outre les opérations requises après le départ d'un nœud de son cluster, afin de mettre à jour la clé de cluster avec les nœuds membres restants.

Lorsqu'un nœud CH $i$ est compromis, la SB génère un message de révocation $E_{k_j^0}(compHeader, i)$ crypté comportant l'identité de ce dernier et l'envoie à tous les nœuds simples $j$ du cluster associé. La SB envoie également à chaque nœud CH $j$ autre que $i$ un message de révocation crypté. Après réception et décryptage du message par le biais de clés personnelles $k_j^0$ partagées avec la SB, les nœuds récepteurs rejettent les clés associées à $i$. Ensuite, chaque nœud simple concerné $j$ essaie de trouver d'autres nœuds CH voisins pour exécuter les étapes qui lui permettront de rejoindre un autre cluster voisin. Si un nœud simple concerné par cette révocation est incapable de trouver un autre nœud CH voisin, il doit en notifier la SB afin que celle-ci procède à la recherche du nœud CH le plus proche et établisse la connexion entre eux.

### 3.3.2 Protocoles Distribués

Dans ces protocoles, les nœuds coopèrent de deux façons. Dans Chan et al. (2003, 2005), Chao et al. (2013) [100] et Chuang et al. (2010) [102], les nœuds non-compromis votent contre les nœuds compromis, alors que dans Moore et al. (2007) les nœuds non-compromis se réélisent pour former un groupe de nœuds de confiance et ainsi exclure les nœuds compromis du réseau.

***Chan et al. (2003)***

Pour réduire les insuffisances liées au mécanisme présenté par Eschenauer et Gligor dans [91], Chan et al. proposent dans [98], la première solution de révocation basée sur l'approche distribuée. Les auteurs décrivent en premier un mécanisme de distribution de clés. Chaque nœud sélectionne aléatoirement $m$ participants et partage avec chacun une paire unique de clés afin de faciliter l'authentification de nœud à nœud (*node-to-node authentication*). Pour chaque nœud, l'ensemble formé par les $m$ participants représente également l'ensemble des membres votants. Avant le déploiement, à chaque membre votant est assignée une clé de vote.

Afin de permettre la vérification de votes en provenance d'autres participants, chaque nœud est aussi pré-chargé avec $\log m$ valeurs de hachage d'authentification, ainsi que la valeur racine de l'arbre de Merkle (*Merkle tree*) [99]. Après détection d'un nœud malveillant par un membre votant, celui-ci

diffuse dans son voisinage sa clé de vote hachée, ainsi que ses valeurs d'authentification. Après réception de ce vote, chaque membre votant le vérifie en utilisant les valeurs d'authentification reçues, ainsi que la valeur racine de l'arbre pré-chargée. En cas de succès, le vote est enregistré et retransmis par broadcast. Si le nombre de votes reçu atteint le seuil de révocation $t$, les membres votants suppriment tous les liens qu'ils avaient avec le nœud compromis.

***Chan et al. (2005)***

Deux ans plus tard, Chan et al. proposent dans [18] une version améliorée de leur première solution de révocation distribuée, en introduisant l'idée de sessions et de partage secret, et en fournissant une définition précise des propriétés de révocation distribuée. Ils ont divisé la durée de vie du réseau en $s_{total}$ sessions afin d'éviter les votes obsolètes. Ils ont également proposé l'utilisation de l'algorithme du partage de secret, générant pour chaque nœud avant son déploiement $s_{total}$ polynômes aléatoires de degré $t$. Puisque chaque nœud a $m$ participants, il est alors pré-chargé avec $m$ votes représentés chacun par $E_{Mask_{ijs}}(q_{js}(x_{ijs}), x_{ijs})$, où $j$ désigne l'un des participants, $s$ désigne la session et $Mask_{ijs}$ est le masque d'activation donné par $j$ à $i$ pendant la phase d'établissement de la connexion. Un arbre de Merkle est généré avec $q_{js}(x_{ijs}), x_{ijs}$ comme feuilles pour chaque nœud $j$ pour chaque session $s$ [99]. Le nœud stocke également les $log\ m$ valeurs de hachage d'authentification et la valeur de la racine dans chaque arbre de Merkle et $H^2(q_{js})$.

Au cours de la session $s$, si le nœud $i$ détecte un nœud compromis $j$, alors $i$ envoie le vote $q_{js}(x_{ijs}), x_{ijs}$ accompagné des $log\ m$ valeurs de hachage d'authentification. Après réception de ce vote, les autres participants du nœud $j$ calculent une valeur à l'aide des informations reçues et la compare avec la valeur racine stockée dans leur mémoire. S'il y'a égalité, les participants enregistreront le vote et continueront à en recevoir, et à effectuer les vérifications nécessaires. Lorsque l'un des participants du nœud $j$ compte au moins $t$ votes dans sa session actuelle, il est capable de calculer le secret $q_{js}$ en utilisant l'algorithme de partage de secret et de le vérifier en utilisant la valeur $H^2(q_{js})$ pré-chargée. Si les deux valeurs sont égales, il met fin aux liens avec le nœud $j$ et diffuse la valeur de hachage de $q_{js}$ dans le réseau. Sinon, il ignore tous les votes reçus à la présente session. Lorsque d'autres participants reçoivent la valeur de hachage de $q_{js}$, ils la vérifient en utilisant la valeur $H^2(q_{js})$ pré-chargée.

***Moore et al. (2007)***

Dans [27] Moore et al. proposent une approche dans laquelle les nœuds non compromis se réélisent (via des votes positifs) pour former un groupe de confiance, excluant ainsi les nœuds compromis du réseau. Ils présentent alors deux solutions de révocation.

Dans la première solution, pour chaque nœud $i$, le jeton d'accès $access_{i,s}$ est généré par une chaîne de hachage, où s désigne la période de temps de 1 à $s_{total}$. Chaque jeton d'accès $access_{i,s}$ est divisé en $m$ parts selon l'algorithme de partage secret à seuil $(k, m)$, où $k$ est le seuil et $m$ est le nombre de membres votants du nœud $i$. Chaque membre votant $j$ du nœud $i$ reçoit une part d'accès $access_{i,s,j}$ et une valeur d'authentification de fin de chaîne notée $access_{i,0}$. Un arbre de hachage est créé avec la valeur d'authentification de fin de chaîne pour chaque nœud dans le réseau en tant que feuille. Le nœud $i$ stocke $access_{i,0}$, ses propres $log\ n$ valeurs d'authentification de chemin, la valeur de hachage de chaque part $access_{i,s,j}$ et la valeur racine de l'arbre. Afin de voter pour ses voisins, le nœud $i$ stocke également les parts $access_{j,s,i}$.

Pendant le temps $s$, le nœud $i$ demande à ses membres votants des parts ou écoute la révélation de celles-ci par ses membres votants. Si le nœud $i$ reçoit au moins $k$ pièces authentifiées, il reconstruit le jeton d'accès $access_{i,s}$ et le transmet par broadcast avec les valeurs d'authentification aux voisins. Après réception du message, chaque voisin le vérifie en calculant la valeur de la racine et compare le résultat à la valeur de la racine pré-chargée. En cas d'égalité, ils continuent à communiquer avec le nœud $i$. Si les voisins détectent le comportement malveillant du nœud $i$, ils suppriment les parts respectives. Par conséquent, $i$ sera incapable de construire le jeton d'accès et de rejoindre le réseau.

Dans la deuxième solution, afin de réduire l'espace de stockage et la consommation d'énergie, chaque nœud $i$ diffuse une liste de contacts contenant des ID des voisins approuvés pendant le temps $s$. La liste est signée par une clé d'authentification de session, qui sera révélée lors de la prochaine période $s + 1$ et authentifiée par une valeur d'authentification de clé distribuée par $i$ pendant la phase de déploiement.

***Chao et al. (2013)***

Afin d'améliorer la solution proposée dans [18] en termes d'exigence d'espace de stockage, de coût de calcul et de communication, Chao et al. décrivent en 2013 dans [100] un nouveau protocole de révocation distribué utilisant le concept de matrice de Blom [101].

Dans leur solution, pour chaque session, chaque nœud a besoin de stocker $t$ éléments d'une ligne d'une matrice de vote et $t$ éléments d'une colonne d'une matrice publique. Au cours de la phase d'initialisation, pour chaque nœud $i$, une matrice publique $G_i$ et $s_{total}$ matrices privées $R_{i(s)}$ sont générées. Ensuite, la matrice de vote pour chaque session $V_{i(s)}$ est générée à partir de $G_i$ et $R_{i(s)}$. La

matrice publique contient des informations publiquement disponibles pour tous les nœuds du réseau alors que la matrice privée utilisée pour chaque session contient des informations secrètes requises à des fins de vérification de vote. Pour chaque session les nœuds pré-chargent aussi les votes encryptés $E_{Mask_{ji(s)}}(V_{i(s)})$ et les informations de la colonne correspondante de la matrice publique.

Notons que dans $V_{i(s)}$ il existe une ligne désignée par la ligne de partage de secret $v_{i(s)}^{\lambda}$, qui n'est pas renseignée pour aucun nœud ni pour aucune session. Chaque nœud participant $i$ stocke également $H^2(v_{i(s)}^{\lambda})$, représentant la valeur de hachage des éléments de la ligne de partage de secret $H(\lambda_{0(s)}||\lambda_{1(s)}||\dots||\lambda_{t-1(s)})$, où $\lambda_{k(s)}$ est le $k$-ème élément de la ligne de partage secret $v_{i(s)}^{\lambda}$. La valeur de hachage $H^2(v_{i(s)}^{\lambda})$ permet aux participants qui ne sont pas des voisins directs du nœud ciblé $i$ de confirmer la validité de la décision de révocation pour le nœud $i$.

Lorsqu'un participant $j$ détecte un nœud malveillant $i$, il diffuse localement, par broadcast, son vote $V_{i(s)}^{j}$ représentant la $j$-ème ligne de la matrice $V_{i(s)}$, et $g_i^j$ représentant la $j$-ème colonne de la matrice $G_i$. À la réception de ce vote, un participant $k$ effectue le processus de vérification en calculant les produits respectifs de $V_{i(s)}^{j}$ et $g_i^k$ et $V_{i(s)}^{k}$ et $g_i^j$, et en les comparant aux valeurs correspondantes dans une matrice qui est le produit de $V_{i(s)}$ et $G_i$. Le processus de vérification d'un vote est illustré schématiquement par la figure 3.5.

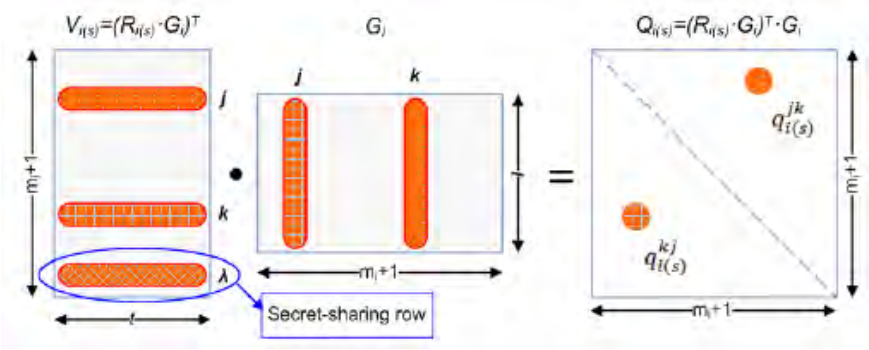



**Figure 3. : Exemple illustratif de vérification de vote.**

Si la vérification réussit, le participant $k$ enregistre le vote. Dans la session en cours, si un participant collecte au moins $t$ votes contre $i$, il calcule les parts de secret à partir des informations reçues et compare le résultat haché avec la valeur stockée dans la mémoire. En cas de succès le participant termine le lien avec le nœud $i$, ajoute $i$ dans une liste de révocation et diffuse la valeur de hachage de $v_{i(s)}^{\lambda}$. Sinon, il ignore toutes les informations associées à la révocation pour le nœud $i$. En recevant le message de révocation, les autres participants le vérifient et en cas de succès suppriment toutes les clés partagées avec le nœud $i$.

***Chuang et al. (2010)***

Au lieu d'une solution basée sur la cryptographie symétrique, Chuang et al. proposent en 2010 dans [102] un protocole de révocation distribué basé sur la clé publique. Leur proposition suppose l'existence une SB qui agit en tant qu'autorité de certification (CA). Pour chaque nœud $i$, SB pré-distribue une clé publique $pk_i$, une clé privée $sk_i$, et émet un certificat de clé publique (PKC), désigné comme $pkc_i$. Lors du déploiement, les nœuds établiront des liens les uns avec les autres en échangeant et en vérifiant les certificats. Lorsqu'un nœud $j$ détecte un nœud compromis $i$, il diffuse par broadcast, un vote ou CRV (*Certificat Révocation Vote*) représenté par $[j, D, revocatioCID \oplus h_k, Timestamp]$, $D$ étant la date où $j$ a reçu la paire de clés publique/privée de SB, $revocationCID$ est l'identifiant de certificat du nœud compromis $i$ et $h_k$ est une valeur de hachage unique dans $pkc_j$. En recevant ce vote, les voisins du nœud $j$ vérifient si le $h_k$ reçu est correct par le biais de $revocatioCID \oplus h_k$. En cas de succès, ils enregistrent le vote. Après obtention de $t$ votes, ils suppriment les clés compromises et diffusent une liste de révocation à tous les membres votants non voisins du nœud compromis.

### 3.3.3 Protocoles Hybrides

Les protocoles de révocation hybrides sont basés sur le concept de regroupement de nœuds, Nous avons pour cette catégorie les travaux de Sanchez et Baldus (2005) [103], Chattopadhyay et Turuk (2012) [106], Jiang et al. (2013) [107] et Ge et Choo (2014) [109].

***Sanchez et Baldus (2005)***

En 2005, Sanchez et Baldus proposent dans [103] une solution de révocation hybride pour les réseaux de capteurs mobiles (*Mobile Sensor Networks (MSNs)*), qui est un sous-ensemble des WSNs. Ils supposent que tous les mécanismes de pré-distribution de clés sécurisés basés sur le pool tels que ceux décrits dans [91], [98], [104], et [105] peuvent être utilisés comme système de distribution de clé associé. Les capteurs forment de manière autonome des MSNs non interconnectés et sont libres de changer de groupes après leur déploiement. Il existe également des SBs mobiles (MSBs), qui se connectent périodiquement aux MSNs. Chaque MSB partage une paire de clés avec chaque nœud dans le réseau et un système de détection d'intrusion est supposé être installé dans chacun de ces nœuds capteurs.

Lorsqu'un nœud compromis est détecté, les capteurs diffusent l'information et suppriment les clés compromises. Si une MSB se connecte au réseau à ce moment-là, le message de révocation collecté localement est envoyé à la MSB. Celle-ci diffuse ce message à d'autres MSBs. Toutes les MSBs, partagent les informations locales de révocation et décident collectivement des nouvelles clés, en remplaçant les clés compromises. Ensuite, chaque MSB se connecte de nouveau au MSN et diffuse

une liste de révocation. Lorsqu'un capteur reçoit la liste il supprime les clés compromises, et entre en contact avec la MSB pour demander de nouvelles clés. Si aucune nouvelle clé n'est laissée dans le pool de clés, une MSB doit informer le serveur d'installation pour mettre à jour le pool.

***Chattopadhyay et Turuk (2012)***

Afin d'améliorer la proposition faite par Chan et al. dans [18], Chattopadhyay et Turuk suggèrent en 2012 dans [106] deux autres protocoles de révocation de clés hybrides nommés *Protocole I* et *Protocole II*. Dans leur papier, les auteurs adressent les insuffisances de [18] et proposent quelques nouvelles astuces pour traiter les problèmes associés.

Les auteurs supposent que le réseau est découpé en parties hexagonales, chacune étant appelée soit une région basique, soit une région non-basique, comme illustré à la figure 3.6.

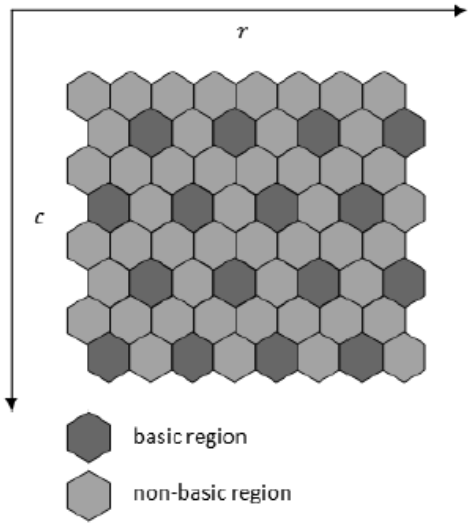



**Figure 3. : Découpage du réseau en régions basiques et non basiques.**

Chaque région basique reçoit un polynôme tri-varié $f_l(x, y, z), 1 \leq l \leq n_b$, $n_b$ étant le nombre de régions basiques, alors que les régions non basiques ont au plus deux polynômes tri-variés, tirés de chacune de ses régions basiques voisines. Le plus haut degré pour ces polynômes tri-variés pour $x, y,$ ou $z$ étant égal à $7k$ où $k$ désigne le nombre maximal de nœuds dans une région.

Chaque nœud stocke des parts du ou des polynômes de sa région et une fonction de hachage unique $h(\,)$. Si $f_l(x, y, z)$ est le polynôme d'une région, alors un nœud ayant un identifiant $j$, aura la part $f_l(j, y, z)$ appelée polynôme d'authentification, $auth_j^l(y, z)$et la part$f_l(x, j, z)$ appelée polynôme de vérification, $verf_j^l(x, z)$. De plus, chaque nœud possède une liste noire et une liste de présumés.

Dans le *Protocole I*, lorsqu'un nœud malveillant $i$ est détecté par le nœud $j$, celui-ci envoie un message $M$, contenant l'ID du nœud $i$, aux nœuds voisins de ce dernier et à la SB. Supposons que le nœud $k$ soit un voisin de $i$. Puisque $j$ et $k$ sont tous les deux voisins de $i$, ils se trouvent soit dans la même région soit dans les régions voisines. Ainsi, ils ont chacun une part d'un polynôme unique $f_l(x, y, z)$. Le nœud $j$ envoie également une valeur de $f_l(j, k, h(M))$ à $k$, avec :

$$f_l(j, k, h(M)) = auth_j^l(k, h(M)) \qquad (3.2)$$

En recevant cette valeur, $k$ calcule $f_l(j, k, h(M))$comme suit :

$$f_l(j, k, h(M)) = verf_k^l\big(j, h(M)\big) \qquad (3.3)$$

et la compare à la valeur reçue. Si elles sont égales, $k$ enregistre le vote et insère l'identifiant du nœud $i$ dans la liste des nœuds suspectés. Lorsque le nœud compte $t$ votes, il supprime toutes les clés partagées avec le nœud $i$ et toutes les clés de chemin via le nœud $i$, et l'ajoute également dans la liste noire. Lorsque la SB reçoit $t$ votes, elle diffuse un message de révocation $M_1$ contenant l'ID et les clés du nœud compromis, ainsi que les polynômes d'authentification $\sum f_l(base.id, y, h(M_1))$ où $base.id$ désigne l'ID de la SB, $l$ est le numéro polynomial de chaque région. Après réception du message de révocation, un nœud $j'$ dans la région $l'$ calcule $f_{l'}(base.id, j', h(M_1))$ et $verf_{j'}^{l'}\big(base.id, h(M_1)\big)$ et compare les deux valeurs. Si elles sont égales, le nœud $j'$supprime toutes les clés associées au nœud compromis et l'ajoute à la liste noire.

Dans le *Protocole II*, les auteurs supposent l'existence d'un nœud de surveillance ou moniteur dans chaque région. L'ensemble du processus est similaire à celui du *Protocole I*, à l'exception de la phase suivante. Après détection d'un nœud compromis, un nœud envoie le vote au nœud moniteur de sa région. Si le nœud compromis se trouve dans une autre région, le nœud moniteur envoie le vote au nœud moniteur de cette région. Sinon, il l'ajoute dans la liste des nœuds suspectés. Lorsque le nombre de votes atteint le seuil $t$, le moniteur correspondant diffuse le message de révocation. Après sa réception, les nœuds le vérifient. Si le message est authentifié, ils suppriment le nœud compromis et l'ajoutent à la liste noire.

***Jiang et al. (2013)***

En 2013, Jiang et al. proposent dans [107] un autre protocole de révocation de clé hybride basé sur le schéma de pré-distribution aléatoire de clé présenté dans [108]. Étant donné que le déploiement du réseau comporte deux niveaux avec le regroupement des capteurs selon la structure de clustering, le protocole de révocation est conçu pour l'intra et l'inter cluster.

Les auteurs supposent que chaque nœud $i$ dans un cluster a $m$ membres votants et doit stocker leur identifiants, $m$ votes et des informations d'authentification, y compris $m$ valeurs racine de hachage et $m\ log_2 m$ valeurs de hachage de chemin frère. Chaque trousseau à clés d'un nœud $i$ contient $m_i$ clés. La durée de vie du réseau est divisée en nombre de périodes notée $\Delta_d$ chacune.

Après la phase de découverte de la clé, le nœud $i$ envoie la clé de vote $k_i$ à ses membres votants. Si le nœud $i$ est compromis et détecté par un autre nœud $j$, ce dernier diffuse son vote $k_j$ et les valeurs de hachage de chemin chiffrées par $k_i$. Après réception de ce vote par les autres membres votants, ils utilisent l'information dans le message pour calculer une valeur avant de comparer le résultat avec la valeur de hachage racine stockée en mémoire. Si ces valeurs sont égales le vote et accepté. Durant la période $\Delta_d$, si $t$ votes son enregistrés, les membres votants suppriment les liens avec $i$ et informent le CH. Sinon, la révocation échoue. Après réception de ce message, le CH diffuse un seul message de révocation dans les sept sous-régions hexagonales. Étant donné que chaque nœud partage une clé de cluster publique avec chaque cluster, il vérifie le message en le recevant. Si la vérification est correcte, le nœud termine le lien avec $i$.

En ce qui concerne l'inter-cluster, les CH établissent d'abord des liens de communication sécurisés l'un avec l'autre en fonction de l'arbre de gestion des clés dans lequel le CH est le nœud feuille géré par ses nœuds parents le long du chemin vers la racine. Si un CH est compromis, toutes les clés du chemin de la feuille vers la racine sont mises à jour par le KC (*Key Center)*. Une chaîne de clés est introduite pour authentifier les clés renouvelées. Lorsque le nœud compromis est supprimé de l'arbre, certains nœuds feuilles peuvent perdre la connexion à d'autres nœuds feuilles. Ainsi, ils doivent diffuser leur trousseau de clés pour découvrir de nouveaux chemins de clés.

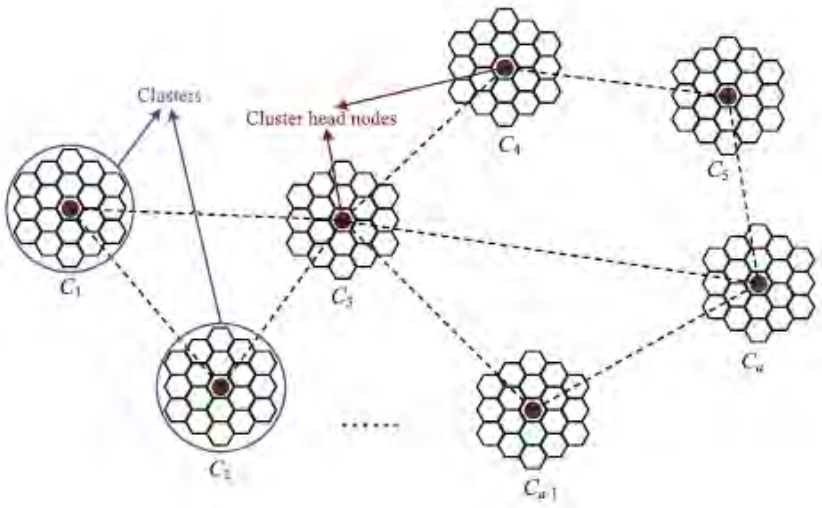



**Figure 3. : Structure en clusters**

***Ge et Choo (2014)***

En 2014, Ge et Choo présentent un nouveau protocole de révocation de clé hybride [109] basé sur [18] qui consiste principalement en une procédure de vote parmi les nœuds et une révocation globale. Dans leur solution, une SB est utilisée pour réaliser la révocation complète. Pour éliminer la contrainte des connaissances d'emplacement avant déploiement, la génération et la distribution autonomes de partage de secret sont réalisées par les nœuds. Chaque nœud $i$ stocke des éléments de son polynôme unique et une valeur de hachage $Hash(s_i)$. Par ailleurs, chaque nœud conserve trois listes. Les deux premières listes contiennent les informations sur les nœuds partenaires, qui partagent une clé de paire avec le nœud $i$. La troisième liste est une liste de révocation, qui est vide initialement.

Par rapport au schéma Chan et al., 2005 [18], ce protocole a ajouté une phase de génération de vote. Dans cette phase, le nœud $j$ stocke $(ID_i, s_{ij}, Mask_i, Hash(s_i))$ dans sa mémoire, où $j$ désigne l'un des participants, $s_{ij}$ est envoyé par le nœud $i$, $Mask_i$ est assigné pendant la phase d'initialisation et doit être utilisé avec d'autres participants pour chiffrer et / ou déchiffrer le vote contre celui-ci. Après cette phase, chaque nœud vote uniquement contre les nœuds avec lesquels il a établi des clés de paire. Dans la session en cours $s$, le nœud $i$ maintient un état d'attente. Une fois qu'un nœud compromis $a$ est détecté, ou le premier vote est reçu, le nœud $i$ change son état en actif et démarre la minuterie. Il va également chiffrer $ID_i$, la part secrète donnée par le nœud $a$, $s_{ai}$ et diffuse ce message. En même temps, il diffuse le vote dans la session en cours et les prochaines sessions. Lorsqu'un autre participant, tel que le nœud $j$, reçoit le vote, il décrypte le message en utilisant le masque correct. Ensuite, le nœud $j$ stocke les paires $(ID_i, s_{ai})$ et rediffuse le message.

Dans sa session actuelle, lorsque le nœud $i$ calcule au moins $t$ parts stockées, il peut calculer le secret $S_a$ en fonction de l'algorithme du partage de secret et la valeur de hachage correspondante. Ensuite, il compare la valeur de hachage avec la $Hash(s_a)$. S'il y'a égalité, le nœud $i$ termine les liens avec le nœud $a$ dans sa liste de liens local et stocke cette information dans la liste de révocation. Ensuite, il diffuse $s_a$ dans le réseau. Pendant ce temps, il informe la SB des informations de révocation.

## Conclusion

Dans la première partie de cette thèse, nous avons donné un aperçu général sur les spécificités des MANETs et des WSNs. Ceci a mis en évidence la difficulté d'appliquer les solutions traditionnelles de gestion des clés à leur l'environnement.

Nous avons aussi introduit la question liée à la révocation de clés dans ces réseaux et montré le rôle primordial qu'elle joue dans la fourniture de solutions de gestion de clés robustes capables

d'adresser les menaces qui pourraient aboutir à la compromission de la confidentialité des clés, la compromission de l'authenticité des clés, l'usage non autorisé de clés.

Nous avons également étudié dans cette première partie les différents protocoles et solutions de révocation de clés proposés pour les MANETs et les WSNs. En raison de l'absence d'infrastructure fiable, pour les MANETs aucune solution n'appartient à la catégorie centralisée, mais malgré les contraintes en ressources des nœuds, les solutions adoptées sont basées sur la cryptographie asymétrique. Par contre en ce qui concerne les WSNs, l'inadaptation de la cryptographie asymétrique a conduit les recherches dans le domaine de la révocation de clés vers la cryptographie symétrique et ainsi vers les méthodes basées sur la pré-distribution de clés.

# Deuxième Partie :

# CONTRIBUTIONS

# Chapitre 4 ANALYSE DES PROTOCOLES DE RÉVOCATION DE CLÉS DANS LES MANETS

Ayant noté l'absence d'études approfondies visant à évaluer et à comparer les différentes propositions de solutions de révocation de clés, nous nous proposons dans la deuxième partie de cette thèse plus précisément dans ce chapitre et dans le chapitre suivant d'analyser les protocoles existants décrits dans le chapitre précédent afin de faire ressortir leurs forces et leurs faiblesses ainsi que les défis qui sont restés non résolus. Pour mener à bien cette analyse nous définissons d'abord un ensemble de critères qui sont utilisés par la suite pour l'évaluation.

## 4.1 Critères d'évaluation des protocoles

Selon les exigences des MANETs et la classification des différents protocoles étudiés, nous avons identifié les critères suivants concernés par l'évaluation des protocoles de révocation :

- **Sécurité**
  - Attaques potentielles ;
  - Services fournis ;
  - Résistance face à ces potentielles attaques.

- **Efficacité**
  - Espace de stockage requis ;
  - Charge de communication ;
  - Charge de calcul ;
  - Délai ou latence de la révocation.

- **Applicabilité**
  - Type de réseau ;
  - Taille du réseau ;
  - Mobilité des nœuds.

### 4.1.1 Sécurité

Afin d'évaluer la sécurité de chaque protocole selon ses forces et vulnérabilités, il est nécessaire de déterminer les attaques potentielles auxquelles il peut faire face ainsi que sa résistance face à ces dernières.

L'exécution d'un protocole de révocation de clés repose essentiellement sur l'échange de messages, donnant l'alerte à propos du comportement d'un nœud et informant à propos de la prise de décision de son exclusion du réseau. Lorsqu'un mécanisme assurant la protection de ces échanges fait défaut dans un protocole de révocation de clés, aucune garantie n'est donnée quant à la bonne exécution des autres mesures de sécurité, mises en place par le mécanisme. Ce faisant, l'authentification des parties communiquant et l'intégrité des données échangées constituent une préoccupation de premier plan pour empêcher aux nœuds non autorisés de recevoir les principaux éléments (par exemple les clés, les certificats) qui peuvent ensuite être utilisés pour prouver le statut de membre légitime du réseau.

Pour garantir la fiabilité et l'efficacité d'un mécanisme de révocation de clés, le système doit aussi être protégé contre les nœuds compromis, identifiés et exclus du réseau suite à la révocation de leurs clés. Pour ce faire, le mécanisme de révocation doit être accompagné d'une solution empêchant leur réinsertion ou réintégration dans le réseau. A ce titre tout autant que les deux services précités, le service de contrôle d'accès au réseau est également requis.

### 4.1.2 Efficacité

Pour garantir leur efficacité, les protocoles de révocation de clés doivent s'exécuter en tenant en compte des contraintes en termes de ressources des nœuds du réseau, mais également réduire au maximum le délai requis pour révoquer un nœud et l'exclure du système. Ce faisant, un mécanisme de révocation de clés doit alors satisfaire aux exigences d'efficacité suivantes [50], [74] :

- **Faible stockage :** la taille occupée par les données requises par les protocoles pour garantir la fiabilité des échanges doit être aussi faible que possible pour s'adapter à la mémoire limitée des nœuds ;
- **Faible coût ou charge de calcul :** la charge de calcul associée au processus de révocation de clés ne doit pas être lourde, car les nœuds possèdent un CPU à faible puissance ;
- **Faible coût ou charge de communication :** la taille globale des données échangées lors du processus de révocation des clés, traduite par le nombre de messages échangés et leur contenu, doit être faible. En effet, le processus ne doit pas introduire de lourdes charges de communication afin d'éviter l'épuisement des réserves en d'énergie des nœuds ;
- **Court délai ou latence de la révocation :** le protocole de révocation de clés doit pouvoir s'exécuter sur une durée assez courte, afin d'empêcher des dommages supplémentaires, causés par la présence de nœuds compromis dans le système.

Pour l'évaluation et la comparaison des différentes solutions, nous évaluons pour chaque protocole, la charge associée au processus de révocation de clés d'un nœud, en déterminant la charge de stockage

et la charge de calcul générée au niveau de chaque nœud participant, mais également la charge de communication et la charge globale de calcul générées par le processus.

### 4.1.3 Applicabilité

Elle se réfère à la possibilité d'une bonne mise en pratique des protocoles de révocation de clés étudiés dans l'environnement des MANETs, en se basant sur les résultats obtenus, suite à l'étude des différents protocoles, l'analyse de leur sécurité et l'analyse de leur performance. Ce faisant, pour ce critère, nous intéressons :

- **au type de réseau** sur lequel le protocole de révocation peut être appliqué (réseau planifié ou auto-organisé (réellement ad hoc)) ;
- **à la taille du réseau,** car pendant toute la durée de vie du réseau, les nœuds peuvent rejoindre ou quitter le réseau. Ainsi les solutions de révocation de clés proposées doivent être adaptables à différentes tailles de réseau en permettant :
  - *i)* d'ajouter en toute simplicité de nouveaux nœuds au système ;
  - *ii)* le passage à l'échelle en maintenant, les caractéristiques de sécurité et d'efficacité ;

- **à la mobilité des nœuds,** car les solutions de révocation de clés proposées doivent s'exécuter en toute sécurité et efficacement en dépit du mouvement des nœuds, qui sont soit censés participer au processus de révocation, soit des adversaires ciblés pour la révocation de leurs clés.

## 4.2 Evaluation de la sécurité des protocoles de révocation de clés dans les MANETs

### 4.2.1 Les protocoles Distribués

#### 4.2.1.1 Attaques de type Man-in-the-middle

La protection des échanges par des mécanismes cryptographiques fournissant les services tels que l'authenticité et l'intégrité procure aux protocoles des moyens de contrer plusieurs attaques de type man-in-the-middle, telles que **l'usurpation d'identité ou impersonation, la modification de messages,** la **fabrication de messages etc**. La plupart des solutions de révocation distribuées proposées pour les MANETs à savoir [6], [8], [10] et [82] préconisent l'utilisation des signatures numériques pour garantir l'authenticité et l'intégrité des messages d'accusation. Cependant, due aux contraintes en ressources des MANETs, le coût associé à l'utilisation de telles opérations est encore sensiblement plus élevé lorsque celui-ci est comparé au coût associé aux opérations cryptographiques

symétriques. Ceci dû au fait que de telles opérations de signature génèrent un coût supplémentaire associé aux opérations arithmétiques telles que l'addition, la multiplication, le modulo.

#### 4.2.1.2 Attaque par exhaustion de batterie

La plupart des protocoles distribués notamment [6], [8], [10] et [82] ont abordé le problème d'authentification de la diffusion (*broadcast authentication*) en utilisant des signatures numériques. Cependant, le coût associé à l'utilisation de ces opérations cryptographiques est encore beaucoup plus élevé que celui des opérations cryptographiques symétriques. En outre, lorsqu'elles sont fréquemment effectuées, elles consomment considérablement la puissance de la batterie des nœuds. Ces inconvénients rendent les protocoles utilisant les signatures vulnérables aux attaques par exhaustion de batterie. Par exemple, un attaquant peut injecter de faux paquets de diffusion pour forcer les nœuds du réseau à exécuter une vérification de signature coûteuse. Cependant la vulnérabilité face à ce genre d'attaque est limitée avec le protocole [82] qui préconise la prise en compte du nombre d'accusations émises par un nœud dans l'évaluation de la fiabilité de ce dernier. Plus explicitement, avec [82], la multiplicité des messages d'accusations envoyés par un nœud entraine l'accroissement de la valeur de la *dishonesty function* $d(x)$ requise pour la prise de décision de révocation.

Les insuffisances associées à l'usage de ces signatures numériques amènent les auteurs de [28], [77], et [79] à proposer des solutions combinant la cryptographie asymétrique avec celle symétrique, afin de générer des clés qui seront utilisées pour produire des MACs (*Message Authentication Code*), garantissant l'authenticité et l'intégrité des messages d'accusation. Toutefois il est à noter que malgré ces améliorations, [77] et [79] présentent toujours une vulnérabilité face aux attaques par exhaustion de batterie. En fait, [77] utilise à la place des signatures numériques une technique d'authentification du broadcast basée sur µTesla, ce qui le maintient toujours enclin aux attaques par exhaustion de batterie [110]. Avec [79], la vulnérabilité à ce type d'attaque est due à la propagation par unicast de messages d'accusation de taille non négligeable vers le voisinage à $m$ sauts.

Il est à noter que le protocole décrit dans [83] ne donne aucune précision quant au mécanisme cryptographique qui sera utilisé pour assurer la protection des communications. Ceci constitue une faille de sécurité majeure puisque l'absence de fourniture de service assurant l'authenticité et d'intégrité des données échangées remet en question toute la sécurité du protocole.

En plus des attaques génériques citées ci-dessus, le protocole de révocation distribué peut être sujet à plusieurs autres attaques. Celles-ci pouvant particulièrement être mises en œuvre par des adversaires qui exercent un contrôle complet sur les nœuds compromis qui, à leur tour, agissent comme des participants actifs. Ces derniers peuvent alors tenter de révoquer sélectivement les nœuds non

compromis en donnant par exemple de fausses alertes. Ils peuvent aussi agir afin d'empêcher ou de bloquer leur propre révocation ou celle des autres nœuds malveillants détectés. Nous notons que, pour renforcer la robustesse du système face à ces potentiels abus, généralement les protocoles de révocation distribués proposés pour les MANETs s'appuient en plus sur un mécanisme de gestion de la confiance distribué. Cependant la gestion de confiance peut elle aussi être une cible attrayante pour les attaquants, car les décisions importantes de révocation peuvent être prises en fonction de la confiance accordée aux nœuds intervenants [111]. Ceci nous amène aussi à prendre en compte dans notre analyse de sécurité les possibles attaques qui peuvent être lancées contre le mécanisme sous-jacent de gestion de la confiance.

#### 4.2.1.3 Attaque par fausse accusation ou Bad Mouth Attack (BMA)

Cette attaque se produit lorsqu'un nœud lance de fausses accusations intentionnellement contre un autre nœud légitime du réseau. Cette attaque est très fréquente dans les méthodes de révocation distribuées. Pour réduire l'impact de cette attaque les auteurs suggèrent la combinaison d'observations multiples pour la prise de décision de révocation. Pour ce faire, le vote avec seuil est utilisé comme première ligne de défense par la quasi-totalité des protocoles distribués excepté celui présenté dans [82]. Ceci leur procure une certaine résistance face aux attaques par fausse accusation, dans la mesure où une seule fausse accusation n'aura pas un aussi grand impact dans la prise de décision de révocation. Cependant cette mesure est loin d'être suffisante pour contrer ou protéger le système contre cette menace, puisque les nœuds attaquants peuvent monter une attaque sybil et /ou se coaliser pour contourner le seuil de révocation établi et réussir la révocation d'un nœud légitime, d'où le besoin de mécanismes de protection supplémentaires pour éliminer les vulnérabilités des protocoles distribués face aux attaques par fausse accusation.

Le protocole Chaib et al. Dans [82], bien que n'utilisant pas une approche de vote avec seuil propose également une solution qui combine des observations multiples avant la prise d'une décision de révocation. Afin de combattre les attaques par fausse accusation, [82] se base sur une fonction $d(x)$ nommée *dishonesty function*, utilisée de sorte à considérer le nombre d'accusations portées contre le nœud accusateur et le nœud accusé, mais aussi le nombre d'accusation lancées par le nœud accusateur et le nœud accusé.

#### 4.2.1.4 Attaque sybil, attaques par coalition de nœuds

Comparée aux autres, l'approche entièrement distribuée utilisée avec ces protocoles les rend encore plus vulnérables aux attaques sybil et aux attaques par coalition de nœuds. Notons qu'en ce qui concerne la révocation distribuée, les adversaires peuvent monter une attaque sybil afin de pouvoir monter par la suite une attaque par coalition de nœuds physiques et/ou virtuels. Ces attaques sont mises en œuvre par plusieurs nœuds physiques ou virtuels collaborant, afin *i)* d'orchestrer la

révocation de nœuds légitimes ou *ii)* d'organiser la réinsertion dans le réseau de nœuds malveillants après leur révocation. Notons que le deuxième scénario mentionné ci-dessus n'est possible que dans les cas où le mécanisme de contrôle d'accès au réseau est assuré de façon auto-organisée par les nœuds du réseau comme c'est le cas dans [6] et [8] sans l'aide ou l'implication d'une tierce partie de confiance *off-line*.

Les premières propositions de solutions de révocation distribuées adoptant l'approche de vote simple basée sur le seuil, notamment [6], [8] et [10] présentent plus de vulnérabilités face aux attaques sybil et/ou aux attaques par coalition de nœuds physiques ou virtuels. Les nœuds coalisants lancent conjointement de fausses accusations contre les nœuds normaux afin de contourner le seuil de révocation défini. Ces derniers réussiront facilement à révoquer un nœud légitime et à l'exclure du réseau dès que leur nombre atteint la valeur du nombre seuil. Notons qu'avec [10], il est plus difficile de réussir une telle attaque si le nombre de D-PKGs (*Distributed Private Key Generators*) pré assignés au nœud ciblé est inférieur au seuil de révocation fixé. Dans ce cas afin de pouvoir construire un message de révocation complet correct, le D-PKG leader a non seulement besoin de coaliser avec les D-PKGs pré assignés au nœud cible, mais doit aussi coaliser avec d'autres D-PKGs extérieurs. Ainsi avec [10], une attaque par coalition de nœud ne peut conduire à la révocation d'un nœud normal que lorsqu'elle est lancée par un groupe de nœuds bien sélectionnés.

Toujours dans le but de renforcer leur résistance face aux attaques par coalition, les propositions plus récentes telles que [28], [77], [79], et [83] proposent des mécanismes de vote pondéré plus complexes qui associent à chaque nœud votant un poids dont l'estimation se base sur d'autres paramètres. [82] combat également les attaques sybil ou attaques par coalition de nœuds avec une approche similaire, bien que non basé sur le vote par seuil, puisque l'historique des accusations du nœud accusateur et celle du nœud accusé permettent l'évaluation de la *dishonesty function* $d(x)$.

Les protocoles [6] et [8] font face à la deuxième forme d'attaque décrite ci-dessus (scénario 2), où les nœuds physiques ou virtuels lancent une attaque sybil et /ou une attaque par coalition pour annuler la révocation d'un nœud par sa réinsertion dans le réseau. En effet dans [6] et [8], les fonctions de contrôle de l'autorité de confiance, qui s'occupe de l'amorçage du système sont distribuées entièrement à tous les nœuds du réseau. A cet effet, la clé secrète du réseau est partagée par tous les nœuds du réseau selon un mécanisme de partage avec seuil $(t, n)$, $n$ etant le nombre total de nœuds dans le réseau. Avec cette approche, l'identification et l'enregistrement de nouveaux nœuds qui se joignent au réseau se fait en ligne, et sont assurés par plusieurs nœuds. Bien que répondant à plusieurs spécificités des MANETs, cette approche rend le système encore plus vulnérable aux attaques sybil et aux attaques par coalition de nœuds, puisqu'avec des identités différentes, les nœuds attaquants pourront collecter autant de parts secrètes nécessaires à la reconstruction de la clé secrète.

Ceci constitue une grande faille qui affectera beaucoup l'efficacité des protocoles de révocation car après exclusion d'un nœud compromis, celui-ci pourra bénéficier de l'aide des nœuds attaquants afin de mettre à jour son jeton d'accès et regagner illégalement le réseau.

Pour résoudre ce problème, les protocoles proposés par la suite notamment [10], [28], [77] et [79] supposent l'existence d'une tierce partie de confiance, qui est soit une CA, soit un KGC *off-line,* qui se charge de vérifier les identités des nœuds avant leur délivrer le certificat et/ou la paire de clés publique/privée. L'application de cette mesure non seulement procure une ligne de défense contre les accès frauduleuse au réseau mais aussi procure une ligne de défense supplémentaire pour prévenir les attaques sybil et/ou attaques par coalition, lancées dans le but de contourner le seuil de révocation défini.

#### 4.2.1.5 Attaques d'itinérance

Avec les protocoles distribués limitant la portée de propagation des accusations vers le voisinage à $m$ sauts, tels que [6], [28] et [79], cette attaque peut être lancée par un nœud compromis détecté afin d'échapper à la sanction. Ce nœud attaquant peut alors se déplacer vers un nouveau voisinage, espérant que les nœuds de cette zone n'ont pas reçu d'accusations lancées contre lui. Pour prévenir cette attaque, [6], [28] et [79], proposent de choisir le paramètre $m$ de sorte que, avant l'expiration de son ticket actuel ou de sa clé publique actuelle, le nœud compromis ne puisse pas sortir de la zone où il est reconnu coupable par les accusations. Une façon de définir $m$ est basée sur la période de validité du ticket ou de la clé publique, la gamme de transmission sans fil et la vitesse de déplacement maximale des nœuds.

Le protocole [82] est par contre vulnérable aux attaques d'itinérance, puisqu'il préconise juste la propagation d'accusations vers le voisinage à un saut. Ce faisant, un adversaire itinérant pourra trouver plus facilement une nouvelle localité où aucune accusation n'est portée contre lui, mais aussi aucune accusation n'est émise par lui. Par conséquent dans son nouveau voisinage, la valeur de sa *dishonesty function* $d(x)$ sera nulle.

**Tableau 4. : Protocoles Distribués – Résumé de l'analyse de sécurité**

| Protocoles Distribués | Services fournis | | Résistance aux potentielles attaques | | | | | | Niveau de sécurité |
|---|---|---|---|---|---|---|---|---|---|
| | Authenticité | Intégrité | MiM | Exhaus. batterie | Fausse accusation | Sybil | Coalition de nœuds | Attaque par itinérance | |
| Luo et al. (2004) [6] | ✓ | ✓ | ✓ | × | × | × | × | ✓ | Faible |
| Saxena et al. (2004) [8] | ✓ | ✓ | ✓ | × | × | × | × | ✓ | Faible |
| Zhang et al. (2006) [10] | ✓ | ✓ | ✓ | × | × | ✓ | ✓ | ✓ | Moyen |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Arboit et al. (2008) [77]** | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | **Convenable** |
| **Fan et Gong (2008) [79]** | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | **Convenable** |
| **Hoeper et Gong (2009) [28]** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **Bon** |

**Tableau 4. : Protocoles Distribués – Résumé de l'analyse de sécurité (suite)**

| **Protocoles Distribués** | **Services fournis** | | **Résistance aux potentielles attaques** | | | | | | **Niveau de sécurité** |
|---|---|---|---|---|---|---|---|---|---|
| | **Authenticité** | **Intégrité** | **MiM** | **Exhaus. batterie** | **Fausse accusation** | **Sybil** | **Coalition de nœuds** | **Attaque par itinérance** | |
| **Chaib et al. (2012) [82]** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | **Convenable** |
| **Guo et al. (2014) [83]** | × | × | × | – – – | ✓ | ✓ | ✓ | ✓ | **Faible** |

✓ : le protocole fournit le service ou résiste à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ =nombre d'attaques auxquelles le protocole résiste / nombre d'attaques potentilles considérées.
Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x < 50\%$, Moyen : $50 \leq r_x < 75\%$, Convenable : $75 \leq r_x < 100\%$, Bon : $r_x = 100\%$.

Le tableau 4.1 nous permet de remarquer que les premières propositions de solutions de révocation distribuées pour les MANETs sont vulnérables à un grand nombre d'attaques potentielles et présentent alors un faible niveau de sécurité. En éliminant l'utilisation de signatures numérique, en renforçant le protocole de révocation par un mécanisme de gestion de la confiance, et en limitant le nombre de retransmissions de messages de révocation, les propositions qui ont suivies trouvent un moyen de contrer les attaques telles que les attaques par exhaustion de batterie, les attaques par fausse accusation les sybil attaques et les attaques par coalition de nœuds. Ce qui permet d'atteindre un niveau convenable de sécurité.

### 4.2.2 Les protocoles décentralisés

Le protocole proposé par Clulow et Moore dans [27] préconise l'usage d'une signature qui est soit basée sur la cryptographie symétrique, soit sur celle asymétrique afin de garantir l'authenticité et l'intégrité de la note de révocation et fournir une défense contre les attaques de type man-in-the-middle.

Avec ce protocole, la décision de révocation revient à un et un seul nœud sans l'implication d'une autorité centralisée ou besoin de coopération entre les nœuds. Ceci protège le système contre les attaques dont souffrent les protocoles ayant recours à la coopération pour la prise de décisions telles que les attaques par coalition, les attaques par fausse accusation, les attaques par exhaustion de batterie

etc. Cependant, cette approche décentralisée ne protège pas pour autant le protocole contre les attaques sybil ou les attaques par réplication de nœuds. En effet, un nœud attaquant peut créer plusieurs fausses identités pour des nœuds virtuels afin de les faire porter la sanction (qui est ici la révocation de clés du nœud accusateur ou son suicide), qui devrait être infligée au nœud accusateur malveillant réel [112], [113]. L'approche distribuée décrite dans [27] ne prévoit aucune mesure pour parer à cette menace, les auteurs supposent juste que des mécanismes sont mis en place pour détecter et prévenir les attaques par réplication de nœuds et les attaques sybil.

Une autre vulnérabilité majeure qu'il faut noter avec cette approche basée sur le suicide est la possibilité de lancer une attaque par révocation [93] par n'importe quel nœud dans le réseau, ce qui constitue une faille importante, qui remet en question la fiabilité des solutions décentralisées.

**Tableau 4. : Protocoles Décentralisés – Résumé de l'analyse de sécurité**

| Protocoles Décentralisés | | Services fournis | | Résistance aux potentielles attaques | | | | | | | Niveau de sécurité |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Authenticité | Intégrité | MiM | Exhaus. batterie | Fausse accusation | Coalition de nœuds | Sybil | Réplication | Attaque par révocation | |
| Moore et al. (2006, 2007) [84], [27] | avec crypto. asym. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | Moyen |
| | avec crypto. sym. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | Moyen |

✓ : le protocole fournit le service ou résiste à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ = nombre d'attaques auxquelles le protocole résiste / nombre d'attaques potentilles considérées.
Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x < 50\%$, Moyen : $50 \leq r_x < 75\%$, Convenable : $75 \leq r_x < 100\%$, Bon : $r_x = 100\%$.

Le tableau 4.2 montre que malgré l'usage du suicide pour contrer les attaques par fausse accusation, les protocoles décentralisés ne parviennent pas pour autant à assurer un niveau convenable de sécurité. Cette mesure les a au contraire rendus beaucoup plus sujet aux sybil attaques, aux attaques par réplication de nœud et même aux attaques par révocation de nœuds légitimes.

### 4.2.3 Les protocoles Hybrides

Les protocoles de révocation hybrides combinent les méthodes centralisées et celles distribuées pour mieux aborder les problèmes de sécurité et de performance relatifs aux autres types de protocoles.

#### 4.2.3.1 Attaques de type Man-in-the-middle, attaques sybil et attaques par réplication

A la différence des protocoles distribués, les protocoles hybrides ne font pas de l'aspect protection des échanges une préoccupation de premier plan, mais se focalisent plutôt sur l'aspect gestion de la confiance, afin de mieux combattre les abus via les fausses accusations et les attaques par coalition de

nœuds. Ainsi aucune précision n'est donnée quant à la solution qui sera adoptée pour sécuriser les communications contre les attaques de type man-in-the-middle.

En plus de grouper les nœuds en clusters, les protocoles hybrides proposent d'impliquer une autorité centrale de confiance *on-line* afin que celle-ci s'occupe de tâches telles que la délivrance de certificats aux nœuds suite à leur déploiement, la prise de décision de révocation, la mise à jour de listes *warning list* (WL), de *Black list* (BL), et la diffusion d'information de révocation à travers tout le réseau (complète notification). Ce qui procure une résistance face aux attaques sybil et aux attaques par réplication de nœuds. En effet le regroupement de nœuds en clusters procure un moyen de contrer les attaques sybil et les attaques par réplication, puisque chaque nœud CH pourra vérifier la légitimité de chaque nœud qui rejoint son cluster.

#### 4.2.3.2 Attaques par exhaustion de batterie

Avec le protocole décrit dans [73], les messages d'accusation *(Accusation Packet (AP))* sont envoyés à la CA et, à la différence des nœuds simples du réseau, cette entité est considérée comme n'ayant pas de contraintes en termes de ressources, d'où une prévention contre les attaques par exhaustion de batterie. Dans [87], au sein d'un cluster, les nœuds CH ont en charge le rôle de déclencher le vote. Ce qui permet :

*i)* de réduire le risque d'attaque par exhaustion de batterie dans le cas où ces nœuds CH ne sont pas contrôlés par des adversaires ;
*ii)* de réduire les distances de communication et ainsi la consommation énergétique liée au vote ;
*iii)* de réduire au niveau des nœuds votants la charge de stockage, la charge de calcul et ainsi la consommation d'énergie grâce à l'utilisation de la cryptographie symétrique pour sécuriser les votes envoyés par les nœuds simples membres de cluster à leur nœud CH.

#### 4.2.3.3 Attaque par fausse accusation ou *Bad Mouth Attack (BMA)*

Avec [73], dès la réception du premier message d'accusation, le nœud accusé est révoqué par la CA et placé dans la BL. Bien que cette approche procure une prompte révocation de la même manière que les protocoles décentralisés et place le nœud accusateur dans la WL, la vulnérabilité du système aux fausses accusations est toujours présente. En effet la menace que représente la présence de ce nœud attaquant dans le système n'est pas éliminée par son placement dans la WL puisqu'une attaque par coalition de nœuds pourra être organisée pour récupérer ce nœud attaquant via un mécanisme de restauration de nœud. Pour combattre cette attaque, les auteurs donnent en outre au nœud CH la possibilité de restaurer tout nœud considéré comme étant faussement accusé, d'où un moyen de contrer une attaque par fausse accusation si le nœud CH concerné n'est pas compromis.

Pour plus de résistance face aux fausses accusations et une révocation avec plus d'exactitude, [87] maintient au sein d'un cluster l'approche basée sur le vote similaire à celle adoptée par les protocoles distribués. A cet effet une seule fausse accusation ne peut permettre la révocation de clés d'un nœud légitime et un nœud détectant le comportement malveillant n'est pas découragé par une sanction consistant en son placement dans la WL.

#### 4.2.3.4 Attaques par coalition de nœuds

Due à l'approche de vote simple adoptée par le mécanisme de restauration de nœud dans [73] il est possible pour les nœuds attaquants de coaliser non pas pour révoquer un nœud légitime, mais pour récupérer un attaquant placé dans la WL. [87] par contre requiert le vote au sein d'un cluster pour la révocation d'un nœud compromis. Pour combattre les éventuelles attaques par coalition, une procédure de vote plus complexe, faisant intervenir plus de paramètres, tels que le poids de chaque nœud votant, le quota de révocation, est utilisée, et le nœud CH est la seule entité disposant de toutes les informations nécessaires à la détermination de la valeur de ces paramètres. Ainsi en présence dans le réseau de nœuds CH fiables, [87] présente une résistance face aux attaques par coalition de nœuds.

Il est à noter qu'avec les protocoles hybrides proposés pour les MANETs, au sein d'un cluster, le nœud CH est la seule entité ayant le privilège de dresser les lignes de défense contre les attaques par fausse accusation et les attaques par coalition de nœuds. Par exemple :

*i)* ce nœud dans [73], doit intervenir auprès de la CA pour annuler la révocation des clés d'un nœud faussement accusé ;

*ii)* dans [87] le nœud CH doit au préalable détecter un nœud membre compromis et lancer en premier un message à ses membres pour amorcer le vote avant que ces derniers ne puissent émettre leur avis concernant la révocation de ce dernier. La décision finale de révocation d'un nœud revient aussi au nœud CH dans [87].

Ceci nous amène à dire que la sécurité des protocoles hybrides dépend fortement de la fiabilité du nœud CH. Cependant ces protocoles ne fournissent aucun mécanisme permettant la surveillance de ce nœud CH et sa révocation en cas de compromission. Pour ces protocoles, les auteurs limitent leur attention juste à la détection et la révocation de nœuds simples membres d'un cluster. De cette analyse nous déduisons qu'avec les protocoles hybrides, le nœud CH constitue un point de défaillance puisqu'en cas de sa compromission, la sécurité du protocole est grandement affectée.

**Tableau 4. : Protocoles Hybrides – Résumé de l'analyse de sécurité**

| Protocoles Hybrides | Services fournis | | Résistance aux potentielles attaques | | | | | | | Niveau de sécurité |
|---|---|---|---|---|---|---|---|---|---|---|
| | Authenticité | Intégrité | MiM | Exhaus. batterie | Fausse accusation | Coalition de nœuds | Sybil | Réplication | Attaque par révocation | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Lui et al. (2013) [73]** | – – – | – – – | – – – | ✓ | ⓥ | × | ⓥ | ⓥ | ⓥ | **Convenable** |
| **Kim (2016) [87]** | – – – | – – – | – – – | ✓ | ⓥ | ⓥ | ⓥ | ⓥ | ⓥ | **Bon** |

– – – : Néant , ✓ : le protocole fournit le service ou résiste à l'attaque, ⓥ : le protocole dépend d'une hypothèse pour résister à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ = nombre d'attaques auxquelles le protocole résiste / nombre d'attaques potentilles considérées. Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x <$ 50%, Moyen : $50 \leq r_x < 75\%$, Convenable : $75\leq r_x < 100\%$, Bon : $r_x = 100\%$.

Le tableau 4.3 ci-dessus nous permet de remarquer qu'avec l'implication d'une autorité centralisée, les protocoles hybrides proposés pour les MANETs parviennent au moins à atteindre un niveau convenable de sécurité. Toutefois nous notons également que la résistance de cette catégorie de protocoles aux éventuelles attaques dépend fortement de l'hypothèse selon laquelle l'autorité impliquée est hors de portée des adversaires, c'est-à-dire n'est pas susceptible d'être compromise.

## 4.3 Evaluation de l'efficacité des protocoles de révocation de clés dans les MANETs

### 4.3.1 Protocoles Distribués

#### 4.3.1.1 Charge de stockage

Etant donné que l'ensemble des protocoles proposés pour les MANETs sont basés sur la cryptographie asymétrique, tous les protocoles distribués requièrent que chaque nœud stocke au moins sa paire de clés publique/privée. En plus de celle-ci, chaque nœud doit stocker les informations relatives aux clés publiques des parties intervenant dans les échanges. En ce qui concerne les protocoles distribués basés sur PKI, chaque nœud doit stocker en plus de la clé publique de tout autre nœud susceptible de communiquer avec lui, le certificat ou le ticket d'accès de ce dernier. Ceci due à l'absence et à la non implication d'une autorité centralisée *on-line.* En ce qui concerne les protocoles non basés sur PKI, les nœuds doivent stocker en mémoire les paramètres système requis pour le calcul et la vérification de la clé publique des autres nœuds.

En plus des informations cryptographiques, les protocoles distribués requièrent pour la quasi-totalité le stockage d'autres informations relatives au mécanisme de gestion de la confiance sur lequel se base le protocole. A cet effet, les identités des nœuds ayant été révoqués sont stockées par la majorité des protocoles (tous exceptés [10] et [82]) dans une forme de liste de révocation. Pour évaluer la fiabilité des nœuds intervenant dans le processus de révocation et prévenir les éventuels abus, d'autres informations doivent être stockées au niveau des nœuds. Le protocole [83] propose le stockage de ces informations dans deux tables distinctes au niveau de chaque nœud. Cependant ceci n'est pas une solution optimale, vue la capacité de stockage limitée des entités du réseau. La solution présentée par [77] a une charge de stockage plus réduite au niveau de chaque nœud puisqu'une seule

table est requise pour stocker ces données. Pour compacter la taille des données requises par le mécanisme de gestion de la confiance, [28] :

*i)* réduit la portée de propagation des messages d'accusation et le nombre d'entrées de tables qui auraient dû stocker les données ;

*ii)* utilise une représentation sous forme matricielle de celles-ci, ce qui engendre un gain en termes d'espace de stockage.

Les protocoles, [6], [8], [10] et [82], bien que basés sur le vote, ne donnent aucune précision quant à la manière de stocker les votes collectés afin de prendre une décision de révocation.

**Tableau 4. : Charge de stockage des Protocoles Distribués – Résumé de l'analyse**

| Protocoles Distribués | Données stockées par chaque nœud participant | | Evaluation de la charge (Complexité) |
|---|---|---|---|
| | Données cryptographiques | Données non cryptographiques | |
| **Luo et al. (2004) [6]** | - *paire de clés pub./ priv.* | - *ticket ou jeton d'acces*<br>- *jeton d'acces des σvoisins directs*<br>- *liste de revocation de tickets*<br>- *votes collectés* | $\geq \boldsymbol{O(\sigma)}$ |
| **Saxena et al. (2004) [8]** | - *paramètres publiques*<br>- *paire de clés pub./ priv.*<br>- *part secrète (validation de membre)* | - *jeton*<br>- *liste de revocation de membres (MRL)*<br>- *votes collectés* | $\geq \boldsymbol{O(1)}$ |
| **Zhang et al. ((2006) [10]** | - *paramètres publiques*<br>- *paire de clés pub./ priv.*<br>- *cles partages avec les $D-PKGs$*<br>- *part secrète de clé (nœud simple)* | – – – | – – – |
| **Arboit et al. (2008) [77]** | - *$(n-1)$ certificats*<br>- *$(n-1)$ valeurs de chaine d'authentification*<br>- *paire de clés pub./ priv.* | - *table de profil avec $(n-1)$ entrées* | $\boldsymbol{O(n)}$ |
| **Fan et Gong (2008) [79]** | - *paramètres publiques*<br>- *paire de clés pub./ priv.*<br>- *Ω clés de paires*<br>- *Ω clés publiques* | - *liste de revocation de clés (matrice de dimension$(n \times 6)$)*<br>- *matrice des observations $(\sigma \times 4)$* | $\boldsymbol{O(n)}$ |
| **Hoeper et Gong (2009) [28]** | - *paramètres publiques*<br>- *paire de clés pub./ priv.*<br>- *σ clés de paires*<br>- *σ clés publiques* | - *liste de revocation de clés (matrice de dimension$(\Omega \times \Omega + 3)$)* | $\boldsymbol{O(\Omega)}$ |
| **Chaib et al. (2012) [82]** | - *paire de clés pub./ priv.*<br>- *σ certificats*<br>- *σ clés publiques* | – – – | – – – |
| **Guo et al. (2014) [83]** | – – – | - *table des accusation avec $(n-1)$ entrées*<br>- *table renseignant sur le status de chaque noeud $(n-1)$ entrées* | $\boldsymbol{O(n)}$ |

D'après le tableau 4.4, la majorité des protocoles distribués proposés pour les MANETs ont une charge de stockage qui dépendant de la taille du réseau $n$ car la complexité associée est $(O(n))$. Ce qui constitue un problème de performance.

#### 4.3.1.2 Charge de communication

Nous remarquons qu'aucun des protocoles distribués basés sur PKI, à savoir, [6], [77] et [82], ne donne de détails à propos du contenu des messages d'accusation. Ce qui empêche l'estimation précise de leur charge de communication. Pour ces protocoles, nous savons juste que le nombre de messages requis pour révoquer un nœud est limité, puisque les messages d'accusation sont diffusés par broadcast, soit vers le voisinage direct [82], soit vers le voisinage à $m$-sauts [6], soit vers tout le réseau [77]. En ce qui concerne les informations requises pour la vérification de ces messages d'accusation ou vote, [6] et [82] précisent que les signatures numériques sont utilisées. Ceci ne requiert que la détention de la clé publique de l'expéditeur, donc aucune charge de communication supplémentaire puisque les nœuds sont supposés détenir les clés publiques des nœuds susceptibles de leur envoyer un message. [77] par contre préconise l'utilisation de MAC (*Message Authentication Code*) à la place des signatures afin de protéger les communications. Cependant après calcul des différents éléments de la chaine de hachage, la dernière valeur de celle-ci doit être envoyée aux autres nœuds du réseau accompagnée de la signature qui la protège.

Contrairement aux protocoles basés sur PKI, les protocoles basés sur l'identité précisent à chaque fois le contenu des messages échangés. [8] inclut juste l'identité du nœud accusé dans le message d'accusation protégé par une signature basée sur l'identité. [10] ajoute à ce contenu un horodatage pour contrer les attaques par rejeu.

[28] et [79] proposent des messages au contenu plus élargi en incluant à ces derniers d'autres informations. En fait dans [79], chaque message d'accusation inclut l'identité du nœud source, ou nœud accusateur, celle du nœud récepteur, la matrice traduisant des observations de nœud source et la valeur de hachage représentant le MAC du message. [28] propose des messages d'accusation au format similaire, cependant l'identité du nœud récepteur n'est pas incluse et est remplacée par un *hopcount* qui permet de déterminer le nombre de sauts restants pour la propagation de l'information. Avec [79] le même message d'accusation doit être envoyé aux $\Omega$ nœuds du voisinage à $m$ sauts du nœud accusateur. Comparé à [79], [28] engendre une charge de communication supérieure puisqu'en plus d'utiliser le vote avec seuil, chacun des $(\Omega - \rho)$ nœuds situés à moins de $m$ sauts par rapport au

nœud accusateur devra, après mise à jour de sa propre liste de révocation de clés envoyer à son tour par unicast l'information à chacun de ses $\sigma$ voisins directs à un saut.

**Tableau 4. 5 : Charge de communication des Protocoles Distribués – Résumé de l'analyse**

| Protocoles Distribués | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Luo et al. (2004) [6]** | *voisinage* à *m sauts* | *vote* | $---$ | *broadcast* | $t_r$ | $t_r . (---)$ | $\geq \boldsymbol{O(t_r)}$ |
| **Saxena et al. (2004) [8]** | *tout le réseau* | *vote* | *ID nœud ciblé* | *broadcast* | $t_r$ | $t_r . IDs$ | $\boldsymbol{O(t_r)}$ |
| **Zhang et al. (2006) [10]** | *dépend position destinataires* | *vote* | - *ID nœud ciblé*<br>- *Horodatage* | *unicast* | $t_r . \beta$ | $t_r . \beta . (IDs + timestamp) + t.IDs + (ID + val_{rev})$ | $\boldsymbol{O(t_r . \beta)}$ |
| | | *premier mess.rév.* | *ID* | *unicast* | $t$ | | |
| | | *dernier mess.rév.* | - *ID*<br>- *valeur de révocation* | *broadcast* | 1 | | |
| **Arboit et al. (2008) [77]** | *tout le réseau* | *vote* | $---$ | *broadcast* | $n_p$ | $n_p . (---)$ | $\geq \boldsymbol{O(n_p)}$ |
| **Fan et Gong ( 2008) [79]** | *voisinage* à *m sauts* | *mess.accusation* | - 2 *IDs*<br>- une matrice<br>- MAC du mess. | *unicast* | $\Omega$ | $\Omega . (2\ IDs + 1\ matrice + 1\ MAC)$ | $\boldsymbol{O(\Omega)}$ |
| **Hoeper et Gong (2009) [28]** | *voisinage* à *m sauts* | *vote* | - 1 *IDs*<br>- *une matrice*<br>- 1 *val num. (hopcount)*<br>- *MAC du mess.* | *unicast* | $\geq t_r . (\Omega - \rho) . \sigma$<br>$\leq \varepsilon . t_r . (\Omega - \rho) . \sigma$ | $\geq t_r . (\Omega - \rho) . \sigma . (IDs + 1\ matrice + hopcount + 1\ MAC)$<br>$\leq \varepsilon . t_r . (\Omega - \rho) . \sigma . (IDs + 1\ matrice + hopcount + 1\ MAC)$ | $\boldsymbol{O(t_r . \Omega)}$ |
| **Chaib et al. ( 2012) [82]** | *voisinage* à 1 *saut* | *mess.accusation* | $---$ | *broadcast* | 1 | $---$ | $\geq \boldsymbol{O(1)}$ |
| **Guo et al. ( 2014) [83]** | *tout le réseau* | *vote* | $---$ | *broadcast* | $n_p$ | $n_p . (---)$ | $\geq \boldsymbol{O(n_p)}$ |

D'après le tableau 4.5 la majorité des protocoles distribués ont une charge de communication qui ne dépend pas du nombre de nœuds dans le réseau $n$, mais du seuil de révocation établi. Cependant tel n'est pas le cas des protocoles basés sur l'identité puisque ceux-ci utilisent le transfert de votes par unicast. Avec cette catégorie de protocoles basés sur l'identité, la charge de communication est plus grande puisqu'elle dépend d'un autre paramètre qui peut être le nombre de nœuds Ω dans le voisinage délimitant la portée de propagation des votes.

#### 4.3.1.3 Charge de calcul

[6] et [82] se basent sur les signatures à clé publique pour sécuriser les messages d'accusation ou votes. Ce faisant, à la réception de ceux-ci, chaque nœud exécute, pour chaque message, une vérification de signature. Puisqu'avec [6] chaque nœud $i$ propage les accusations uniquement vers son voisinage à $m$ sauts, et que dans celui-ci on considère qu'il y'a Ω nœuds, le nœud $i$ inclus, nous aurons alors pour ce protocole une charge globale de calcul par révocation $c_{[6]}$ qui est égale à :

$$c_{[6]} = t_r[sign_{pub} + (\Omega - 1)\ verif_{sign}] \qquad (4.1)$$

Pour [82] qui ne se base pas sur le vote avec seuil, après réception du message d'accusation et vérification de la signature, la *dishonesty function* est évaluée deux fois ; une fois pour le nœud accusateur et une fois pour le nœud accusé.

[83] ne donne aucune précision à propos des opérations cryptographiques qui permettent la fourniture de service d'authentification et d'intégrité des échanges. Le protocole se contente juste de faire savoir que le vote avec seuil est utilisé et qu'après réception d'un message d'accusation le poids du nœud accusateur est évalué avant d'être ajouté à la somme des poids des autres nœuds ayant accusé le nœud ciblé. Ceci dans le but de comparer la somme finale obtenue au quotient de révocation fixé.

Bien que basé sur PKI, [77] utilise les valeurs successives d'une chaine de hachage en tant que clé symétriques. Ce faisant les MAC (*Message Authentication Code*) seront utilisés pour ce protocole à la place des signatures numériques. A la réception d'un message, chaque nœud calcule alors le MAC associé au message avant de le comparer à celui reçu. En cas de succès, le poids du nœud accusateur est évalué et incrémenté à la somme obtenue. Si $n_p$ est le nombre de nœuds dont la somme de leur poids est supérieure ou égale au quotient de révocation, et que les accusations dans [77] sont propagées vers tout le réseau, nous obtenons alors pour ce protocole une charge globale de calcul par révocation de clé $c_{[77]}$ égale à :

$$c_{[77]} = n_p\ .\ [n\ .\ MAC + (n-1).\ calcul_{poids}] \qquad (4.2)$$

Avec [8], étant donné que le vote avec seuil est employé, pour révoquer un nœud compromis, $t_r$ messages d'accusation signés doivent être envoyés par broadcast à tous les autres nœuds du réseau. Dans [10], les messages d'accusation signés sont en plus chiffrés avec les clés symétriques de paire avant d'être envoyés par unicast aux $\beta$ D-PKG pré-assignés au nœud ciblé. Pour compléter le processus, $t$ D-PKG signent conjointement un message de révocation selon une procédure de signature avec seuil avant de l'envoyer vers le PKG leader. Le message de révocation pourra être constitué par ce nœud leader en exécutant une interpolation de Lagrange (résolution d'un système linéaire avec $t$ inconnus). Après obtention du résultat, le PKG leader calcule, en plus, deux opérations de couplage (*pairing*) [114-116] à des fins de vérification du résultat obtenu avant de diffuser à travers tout le réseau le message complet de révocation contenant l'identité du nœud compromis et la valeur

complète de révocation ainsi calculée. Après réception de ce dernier message, chaque nœud pourra le vérifier en exécutant deux opérations de couplage (*pairing*).

Les protocoles [28] et [79] préconisent l'usage de MACs (*Message Authentication Code*) à la place des signatures basées sur l'identité afin d'assurer la protection des messages. Pour ces deux protocoles, les accusations sont propagées vers le voisinage à $m$-sauts. Dans [79] le vote avec seuil n'est pas employé et chaque nœud doit envoyer par unicast son message d'accusation à tous ses voisins à $m$-sauts. En ce qui concerne les opérations cryptographiques, [28] produit un coût plus élevé comparé à [79]. En effet, avec [28], le vote avec seuil est utilisé. De plus, chaque nœud dans le voisinage à $m$-sauts situé à un nombre de sauts inférieur à $m$ par rapport au nœud accusateur propagera après mise à jour de sa propre liste de révocation l'information vers son voisinage à un saut. Pour ce protocole, dans le meilleur des cas, c'est-à-dire, celui ou le nœud ciblé est révoqué juste après que $t_r$ accusations aient été reçues, le coût de calcul associé aux opérations cryptographiques $c_{[28]}$ est de :

$$c_{[28]} = t_r[\ (\Omega - \rho).2.\sigma.MAC + \rho.MAC] \qquad (4.3)$$

$\rho$ étant le nombre de nœuds situés à $m$ sauts par rapport au nœud accusateur et $\sigma$ le nombre moyen de nœuds dans le voisinage à un saut de chaque nœud du réseau. Dans le pire des cas, le nœud ciblé sera révoqué uniquement après réception d'accusations rapportées, c'est-à-dire les accusations reçues à partir de nœuds du voisinage à un saut reportant des accusations. Dans le cas où toutes les accusations sont rapportées, le vote majoritaire est utilisé et le seuil requis est ε [3], [28]. Ce qui nous donne un coût de calcul $c'_{[28]}$ égal à :

$$c'_{[28]} = \varepsilon.t_r[\ (\Omega - \rho).2.\sigma.MAC + \rho.MAC] \qquad (4.4)$$

En plus du coût associé aux opérations cryptographiques, nous avons en ce qui concerne [79] un coût supplémentaire de calcul, engendré par le mécanisme de gestion de la confiance qui accompagne le protocole. En effet pour contrer les abus, le protocole ajoute à la vérification cryptographique deux autres lignes de défense. D'abord un test de déviation, qui filtre les potentielles fausses accusations, est exécuté, ensuite la théorie de croyance de *Dempster Shafer* [81] est utilisée pour intégrer les rapports des autres nœuds. Le test de déviation consiste en un calcul de quatre probabilités suivant la distribution de Dirichlet. Avec la théorie de *Dempster Shafer*, le nœud récepteur du message d'accusation détermine le poids du nœud accusateur avant de l'utiliser pour la détermination des éléments de la ligne qui correspond au nœud ciblé dans sa propre matrice donnant le statut de nœuds (*Node Status Matrix*).

**Tableau 4. 6 : Charge de calcul des Protocoles Distribués – Résumé de l'analyse**

| Protocoles Distribués | Activité | Nature Opération (s) | Nombre d'opérations / nœud | Charge / activité / nœud (complexité) | Nombre de nœuds participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Luo et al. (2004) [6]** | $vote$ | $signature$ | 1 | $\boldsymbol{O(1)}$ | $t_r$ | $t_r.[sign + \Omega.verif_{sign}]$ | $\boldsymbol{O(\Omega)}$ |
| | $vérif.vote$ | $vérif.sign.$ | $t_r$ | $\boldsymbol{O(t_r)}$ | $\Omega$ | | |
| **Saxena et al. (2004) [8]** | $vote$ | $signature$ | 1 | $\boldsymbol{O(1)}$ | $t_r$ | $t_r.[sign + (n-1).verif_{sign}]$ | $\boldsymbol{O(n)}$ |
| | $vérif.vote$ | $vérif.sign.$ | $t_r$ | $\boldsymbol{O(t_r)}$ | $n-1$ | | |
| **Zhang et al. (2006) [10]** | $vote$ | - $signature$<br>- $chiffrement$ | $\beta$ | $\boldsymbol{O(\beta)}$ | $t_r$ | $\beta.t_r.[sign + chiff + dechiff + verif_{sign}] + t.sign_{part} + 1(resol.syst.lin.) + n.2.(couplage\ (pairing))$ | $\boldsymbol{O(n)}$ |
| | $vérif.vote$ | - $déchiffrement$<br>- $vérif.sign.$ | $t_r$ | $\boldsymbol{O(t_r)}$ | $\beta$ | | |
| | $mess.rév.$ | $signature\ part.$ | 1 | $\boldsymbol{O(1)}$ | $t$ | | |
| | $vérif.mess.rév.$ | - $résol.syst.lin.$<br>- $couplage\ (pairing)$ | 1 | $\boldsymbol{O(1)}$ | 1 | | |
| | $vérif.mess.rév.(suite)$ | $couplage\ (pairing)$ | 1 | $\boldsymbol{O(1)}$ | $n-1$ | | |
| **Arboit et al. (2008) [77]** | $vote$ | - $MAC$ | 1 | $\boldsymbol{O(1)}$ | $n_p$ | $n_p.[n.MAC + (n-1).calcul_{poids}]$ | $\boldsymbol{O(n)}$ |
| | $vérif.vote$ | - $MAC$<br>- $calcul\ poids nœud$ | $n_p$ | $\boldsymbol{O(n_p)}$ | $n-1$ | | |
| **Fan et Gong (2008) [79]** | $vote$ | - $MAC$ | 1 | $\boldsymbol{O(1)}$ | 1 | $\Omega.[2.MAC + 4.calcul_{prob} + .calcul_{poids}]$ | $\boldsymbol{O(\Omega)}$ |
| | $vérif.vote$ | - $MAC$<br>- $calcul\ prob.$<br>- $calcul\ poids nœud$ | 1 | $\boldsymbol{O(1)}$ | $\Omega$ | | |
| **Hoeper et Gong (2009) [28]** | $vote$ | $MAC$ | 1 | $\boldsymbol{O(1)}$ | 1 | $\geq t_r[(\Omega-\rho).2.\sigma.MAC + \rho.MAC]$<br>$\leq \varepsilon.t_r[(\Omega-\rho).2.\sigma.MAC + \rho.MAC]$ | $\boldsymbol{O(\Omega)}$ |
| | $vérif.vote$ | $MAC$ | $entret_r ett_r.\varepsilon$ | $\boldsymbol{O(t_r)}$ | $\Omega$ | | |
| **Chaib et al. (2012) [82]** | $vote$ | $signature$ | 1 | $\boldsymbol{O(1)}$ | 1 | $sign + \sigma.(verif_{sign} + calcul_{fonction})$ | $\boldsymbol{O(\sigma)}$ |
| | $vérif.vote$ | - $vérif.sign.$<br>- $calcul\ val.\ \mathrm{d}(x)$ | 1 | $\boldsymbol{O(1)}$ | $\sigma$ | | |
| **Guo et al. (2014) [83]** | $vote$ | $---$ | $---$ | $---$ | $n_p$ | $---$ | $\geq \boldsymbol{O(n)}$ |
| | $vérif.vote$ | - $---$<br>- $calcul\ poids$ | $n_p$ | $---$ | $n-1$ | | |

Le tableau 4.6 nous fait remarquer que pour presque tous les protocoles distribués basés sur le vote, la charge de calcul produite par nœud est fonction du seuil de révocation établi puisqu'avec ceux-ci nous avons une complexité égale à $O(t_r)$. Le tableau nous fait également noter que la charge globale de calcul générée par le processus de révocation d'un nœud est plus élevée avec les protocoles utilisant la complète notification car avec ceux-ci nous avons $O(n)$.

#### 4.3.1.4 Délai ou latence de la révocation

Tous les protocoles basés sur PKI, ainsi que le protocole basé sur l'identité [8], utilisent le broadcast pour propager les messages d'accusation, soit vers le voisinage direct [82], soit vers le voisinage à $m$ sauts [6], soit vers tout le réseau [8], [77] et [83]. Puisque [82] utilise les signatures pour authentifier les messages d'accusation, et que le vote avec seuil n'est pas employé par ce protocole, le délai de la révocation $T_{[82]}$ sera égal à :

$$T_{[82]} = T_{sign} + T_{transmit} + T_{verif-sign} + c \qquad (4.5)$$

$c$ étant le délai pour l'exécution des opérations non cryptographiques (calcul de deux *dishonesty functions*).

Puisque aucune opération non cryptographique n'est exécutée par les protocoles [6] et [8], et que le vote avec seuil est utilisé, les délais respectifs associés $T_{[6]}$ et $T_{[8]}$ seront alors égal à :

$$T_{[6]} = t_r \,.\, \left[T_{sign} + m.\,(T_{transmit} + T_{verif-sign})\right] \qquad (4.6)$$

$$T_{[8]} = t_r \,.\, \left[T_{sign} + k.\,(T_{transmit} + T_{verif-sign})\right] \qquad (4.7)$$

Avec [77], les MACs (*Message Authentication Code*) sont utilisés à la place des signatures pour protéger les messages d'accusation. Avec ce protocole, le quotient de révocation obtenu dépend de la fiabilité des nœuds votants. Ainsi si $n_p$ est le nombre de nœuds, dont la somme des accusations pondérées est supérieure ou égale au quotient de révocation, nous obtenons un délai $T_{[77]}$ égal à :

$$T_{[77]} = n_p.\,[T_{MAC} + \; k.\,(T_{transmit} + T_{MAC} + c)] \qquad (4.8)$$

[83] ne donne pas de précision concernant les opérations cryptographiques à effectuer pour vérifier les messages d'accusation, mais décrit une approche similaire à celle de [77], propageant les votes pondérés par broadcast à travers tout le réseau. Ce qui produit pour ce protocole un délai de révocation supérieur ou égal à celui de [77].

Les autres protocoles basés sur l'identité [10], [28] et [79] utilisent le transfert par unicast pour la propagation de messages d'accusation. Avec [10], le vote avec seuil est employé et chaque vote signé

puis chiffré doit être envoyé aux $\beta$ D-PKGs assignés au nœud ciblé. Dans le meilleur des cas, ces $\beta$ D-PKGs sont tous situés au voisinage direct des $t_r$ nœuds votants et dans le pire des cas, ces $\beta$ D-PKGs sont tous situés à $k$ sauts de chacun des $t_r$ nœuds votants. Après collecte de $t_r$ votes, un premier message de révocation protegé par une signature partielle est envoyé par $t$ D-PKGs collaborant au D-PKG leader. Dans le meilleur des cas, ces $t$ D-PKGs sont tous situés dans le voisinage direct du D-PKG leader et dans le pire des cas, ils seront tous situés à $k$ sauts du nœud leader. Après réception de ce premier message de révocation et l'exécution de deux opérations de couplage *(pairing)*, le nœud leader diffuse l'information par broadcast à travers tout le réseau.

[79] propage les messages d'accusation protégés par MAC vers le voisinage à $m$ sauts. Avec ce protocole, le nœud accusateur doit envoyer à chaque nœud du voisinage à $m$ sauts le même message. Si Ω est le nombre total de nœuds dans ce voisinage, et $n_i$ le nombre de nœuds situés à $i$ sauts par rapport au nœud accusateur, nous obtenons un délai de révocation $T_{[79]}$ égal à :

$$T_{[79]} = 2.\Omega\,.T_{MAC} + T_{transmit}.\sum_{i=1}^{m} i.n_i + c \qquad (4.9)$$

$c$ étant le délai requis pour l'exécution des opérations non cryptographiques (relatives à la gestion de la confiance).

[28] propage également les messages d'accusation vers le voisinage à $m$ sauts. Cependant, avec ce dernier protocole, après réception d'un message et mise à jour de sa liste de révocation le nœud voisin doit transmettre à ses voisins à un saut cette nouvelle liste si ces derniers appartiennent au voisinage à $m$ sauts du nœud accusateur. Soit $\sigma$ le nombre moyen de nœuds dans le voisinage à un saut de chaque nœud chargé de propager l'accusations. Nous obtenons alors pour la propagation d'une accusation dans [28] un délai $T_{[28]}$ égal à :

$$T_{[28]} = m\,.\sigma.[2.T_{MAC} + T_{transmit}] + c \qquad (4.10)$$

Dans le meilleur des cas, juste $t_r$ accusations seront requises pour la révocation d'un nœud, et dans le pire des cas, $\varepsilon.t_r$ accusations seront requises pour la révocation d'un nœud compromis, $\varepsilon$ étant le seuil requis pour le vote majoritaire.

**Tableau 4. 7 : Délai de révocation des Protocoles Distribués – Résumé de l'analyse**

| Protocoles Distribués | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authen. | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|
| **Luo et al. (2004) [6]** | $mess.accusation$ | $voisinage$ à$msauts$ | $broadcast$ | - $signature$<br>- $vérif.sign$ | $t_r . [T_{sign} + m . (T_{transmit} + T_{verif-sign})]$ | $\boldsymbol{O(t_r . m)}$ |
| **Saxena et al. (2004) [8]** | $mess.\ accusation$ | $tout\ le\ réseau$ | $broadcast$ | - $signature$<br>- $vérif.sign.$ | $t_r . [T_{sign} + k . (T_{transmit} + T_{verif-sign})]$ | $\boldsymbol{O(t_r . k)}$ |
| **Zhang et al. (2016) [10]** | $vote$ | $dépend\ position\ des\ D-PKGs$ | $unicast$ | - $signature$<br>- $chiffrement$<br>- $déchiffrement$<br>- $vérif.sign.$ | $\geq t_r . \beta . [T_{sign} + T_{crypt} + T_{transmit} + T_{decrypt} + T_{verif-sign}]$<br>$\leq t_r . \beta . [T_{sign} + T_{crypt} + k . T_{transmit} + T_{decrypt} + T_{verif-sign}]$ | $\geq \boldsymbol{O(t_r . \beta)}$<br>$\leq \boldsymbol{O(k . t_r . \beta)}$ |
| | $message\ de\ revocation\ vers\ le\ D-PKG\ leader\ envoyé\ en\ meme\ temps\ par\ les\ D-PKGs$ | $dépend\ position\ du\ leader$ | $unicast$ | - $sign. Partielle$<br>- $deux\ bilin. map.$ | $\geq (T_{sign-part} + T_{transmit} + 2 . T_{bilin-map})$<br>$\leq (T_{sign-part} + k . T_{transmit} + 2 . T_{bilin-map})$ | |
| | $message\ de\ revocation\ (notification)$ | $tout\ le\ réseau$ | $broadcast$ | - $deux\ bilin. map.$ | $k . (T_{transmit} + 2 . T_{bilin-map})$ | |
| **Arboit et al. (2008) [77]** | $mess.accusation$ | $tout\ le\ réseau$ | $broadcast$ | - $MAC$ | $n_p . [T_{MAC} + k . (T_{transmit} + T_{MAC} + c)]$ | $\boldsymbol{O(n_p . k)}$ |
| **Fan et Gong (2008) [79]** | $mess.accusation$ | $voisinage$ à$msauts$ | $unicast$ | - $MAC$ | $2 . \Omega . T_{MAC} + T_{transmit} \cdot \sum_{i=1}^{m} i . n_i + c$ | $\boldsymbol{O(\Omega)}$ |
| **Hoeper et Gong (2009) [28]** | $mess.accusation$ | $voisinage\ à\ m\ sauts$ | $unicast$ | - $MAC$ | $\geq t_r . m . \sigma . [2 . T_{MAC} + T_{transmit}] + c$<br>$\leq t_r . \varepsilon . m . \sigma . [2 . T_{MAC} + T_{transmit}] + c$ | $\boldsymbol{O(t_r . m . \sigma)}$ |
| **Chaib et al. (2012) [82]** | $mess.accusation$ | $voisinage\ à\ un\ saut$ | $broadcast$ | - $signature$<br>- $vérif.sign.$ | $T_{sign} + T_{transmit} + T_{verif-sign} + c$ | $\boldsymbol{O(1)}$ |
| **Guo et al. (2014) [83]** | $mess.accusation$ | $tout\ le\ réseau$ | $broadcast$ | - $---$<br>- $calcul\ poids$ | $---$ | $\geq \boldsymbol{O(n_p . k)}$ |

Le tableau 4.7 montre que les protocoles distribués utilisant le transfert de messages par broadcast engendrent un délai qui dépend juste de deux paramètres à savoir le seuil de révocation et le nombre de sauts à parcourir (complexité égale à $O(t_r . k)$ ou $O(n_p . k)$ ou $O(t_r . m)$). Cependant avec celui utilisant le vote et le transfert par unicast notamment [28], le délai de révocation devient plus long puisque dépend également du nombre de nœuds destinataires (complexité égale à $O(t_r . m . \sigma)$).

### 4.3.2 Protocoles Décentralisés

#### 4.3.2.1 Charge de stockage

Avec l'approche basée sur la cryptographie asymétrique, chaque nœud doit stocker sa paire de clés publique/privée, les clés publiques des nœuds avec qui il communique, ainsi que les certificats qui leur correspondent. Avec la cryptographie symétrique, chaque nœud a besoin de stocker la clé de signature privée, la clé publique associée, la valeur racine d'authentification et les $logn$ valeurs de hachage d'authentification de chemin.

**Tableau 4. 8 : Charge de stockage des Protocoles Décentralisés – Résumé de l'analyse**

| Protocoles Décentralisés | | Données stockées par chaque nœud participant | Evaluation de la charge (Complexité) |
|---|---|---|---|
| Moore et al. (2006, 2007) [84], [27] | avec crypto. asym. | - *paire de clés pub./ priv.*<br>- *clés publiques des* $(n-1)$ *autres noeuds du réseau*<br>- *certificats des* $(n-1)$ *autres noeuds du réseau* | $\boldsymbol{O(n)}$ |
| | avec crypto. sym. | - *paire de clés pub./ priv.*<br>- $logn + 1$ *valeurs d'authentification* | $\boldsymbol{O(log\ n)}$ |

Le tableau 4.8 montre que les protocoles décentralisés produisent une charge de stockage qui dépend du nombre de nœuds dans le réseau $n$. Ce qui affecte leur performance avec l'accroissement du nombre de nœuds dans le réseau. L'approche utilisant la cryptographie symétrique améliore cette performance en préconisant le pré chargement de valeurs de hachage. Ce qui réduit la complexité de la charge de stockage pour la ramener à $O(log\ n)$.

#### 4.3.2.2 Charge de communication

Les deux approches proposées par Clulow et Moore dans [27] ont juste besoin d'une transmission par broadcast vers tout le réseau pour transmettre l'information de révocation. Dans celle basée sur la cryptographie asymétrique, le message envoyé par broadcast inclut juste la note de suicide ainsi que le certificat de clé publique. Par contre dans l'autre approche, le message inclura en plus de la note de suicide signée, les $logn$ valeurs d'authentification de chemin.

**Tableau 4. 9 : Charge de communication des Protocoles Décentralisés – Résumé de l'analyse**

| Protocoles Décentralisés | | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| Moore et al. (2006, 2007) [84], [27] | avec crypto. asym. | *tout le réseau* | *mess. suicide* | - *note de suicide*<br>- *certificat* | *broadcast* | 1 | $Note_{suicide} + Certificat$ | $\boldsymbol{O(1)}$ |
| | avec crypto. sym. | *tout le réseau* | *mess. suicide* | - *note de suicide*<br>- *lognval. d'authen.* | *broadcast* | 1 | $Note_{suicide}$ $+ logn\ (valeurs\ de\ hach.)$ | $\boldsymbol{O(logn)}$ |

D'après le tableau 4.9, le protocole décentralisé basé sur la cryptographie asymétrique ne produit aucune charge de communication car l'évaluation de la complexité associée nous permet d'obtenir

$O(1)$. Ceci n'est pas le cas du protocole basé sur la cryptographie symétrique puisque le message doit en outre inclure les $logn$ valeurs de hachage.

#### 4.3.2.3 Charge de calcul

En utilisant l'approche basée sur la cryptographie asymétrique, chaque nœud vérifiera la note signée reçue. Avec l'approche symétrique, avant de vérifier la signature de la note de suicide, $log\ n + 1$ opérations de hachage doivent être calculées afin de pouvoir vérifier l'authenticité de la clé publique du nœud expéditeur de cette dernière.

**Tableau 4. 10 : Charge de calcul des Protocoles Décentralisés – Résumé de l'analyse**

| Protocoles Décentralisés | | Activité | Nature Opération(s) | Nombre d'opérations / nœud | Charge / activité / nœud (complexité) | Nombre de nœuds participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| Moore et al. (2006, 2007) [84], [27] | avec crypto. asym. | $mess.\ suicide$ | $signature$ | 1 | $\boldsymbol{O(1)}$ | 1 | $1\ sign + (n-1).verif_{sign}$ | $\boldsymbol{O(n)}$ |
| | | $vérif.\ mess.$ | $vérif.sign.$ | 1 | $\boldsymbol{O(1)}$ | $n-1$ | | |
| | avec crypto. sym. | $mess.\ suicide$ | $signature$ | 1 | $\boldsymbol{O(1)}$ | 1 | $1\ sign + (n-1).(logn+1)hachages + (n-1).verif_{sign}$ | $\boldsymbol{O(n.logn)}$ |
| | | $vérif.\ mess.$ | - $hachage$ - $vérif.sign.$ | - $logn+1$ - 1 | $\boldsymbol{O(logn)}$ | $n-1$ | | |

D'après le tableau 4.10, seul le protocole décentralisé utilisant la cryptographie symétrique génère par nœud une charge de calcul puisqu'avec celui-ci la complexité associée est de $O(logn)$. Par contre pour les deux types de protocoles rencontrés avec cette catégorie décentralisée, puisque tous les nœuds doivent intervenir dans le processus de révocation d'un autre nœud, la charge globale générée par le processus de révocation d'un nœud est dépendante du nombre de nœuds dans réseau (complexité égale à $O(n)$ ou $O(n.logn)$).

#### 4.3.2.4 Délai ou latence de la révocation

Avec les deux approches, la note de suicide signée est transmise par broadcast à l'ensemble des nœuds du réseau. L'approche utilisant la cryptographie asymétrique requiert, pour les opérations d'authentification, l'exécution d'une opération de signature par le nœud expéditeur et la vérification de cette dernière par les nœuds récepteurs de la note. L'approche utilisant la cryptographie symétrique exécutera les mêmes opérations afin de vérifier l'authenticité et l'intégrité de la note de suicide. Cependant pour cette deuxième méthode, afin de vérifier l'authenticité de la clé publique de l'expéditeur du message, les nœuds récepteurs devront, en plus, exécuter $logn + 1$ opérations de hachage.

**Tableau 4. 11 : Délai de révocation des Protocoles Décentralisés – Résumé de l'analyse**

| Protocoles Décentralisés | | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authen. | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|---|
| Moore et al . (2006, 2007) [84], [27] | avec crypto. asym. | $mess.\ révocation$ | $tout\ le\ réseau$ | $broadcast$ | $-signature$ $-\ vérif.sign.$ | $T_{sign} + k.(T_{transmit} + T_{verif-sign})$ | $\boldsymbol{O(k)}$ |
| | avec crypto. sym. | $mess.\ révocation$ | $tout\ le\ réseau$ | $broadcast$ | $-\ signature$ $-\ hachage$ $-\ vérif.sign.$ | $T_{sign} + k.(T_{transmit} +$ $(logn + 1).\ T_{hash} + T_{verif-sign})$ | $\boldsymbol{O(k.logn)}$ |

A travers le tableau 4.11, nous notons que le protocole décentralisé utilisant l'approche basée sur la cryptographie symétrique produit un délai de révocation plus long dû au calcul des $logn$ valeurs de hachage pour la vérification de l'authenticité du message.

### 4.3.3 Protocoles Hybrides

#### 4.3.3.1 Charge de stockage

Bien que basés sur PKI, les données et opérations cryptographiques utilisées dans [73] et [87] ne sont pas spécifiées. Pour ces protocoles, l'attention est principalement portée sur l'aspect gestion de la confiance. A cet effet, pour [73], chaque nœud stocke deux listes locales à savoir la WL (Warning List) et la BL (Black List) permettant de déterminer respectivement la fiabilité et le statut de nœuds membres du même cluster.

Dans [87], le nœud CH doit stocker pour chaque nœud membre de cluster l'historique de ses accusations et de celles portées contre lui afin de pouvoir déterminer le poids de chaque nœud intervenant dans le processus de révocation de clé. Cependant aucune précision n'est donnée quant à la manière ou le format qui sera utilisé pour stocker ces informations. Même si les nœuds CH sont choisis comme étant ceux qui ont plus de capacité de stockage, la gestion des données pré-mentionnées peut entrainer une consommation non négligeable de l'espace mémoire disponible et ceci s'accentuera avec l'accroissement du nombre de nœuds dans le cluster.

**Tableau 4. 12 : Charge de stockage des Protocoles Hybrides – Résumé de l'analyse**

| Protocoles Hybrides | Données stockées par chaque nœud participant | Evaluation de la charge |
|---|---|---|

| | Données cryptographiques | Données non cryptographiques | (Complexité) |
|---|---|---|---|
| **Lui et al. (2013) [73]** | – – – | $-Warning\ list\ (WL)$<br>$-\ Black\ list\ (BL)$ | $\geq \boldsymbol{O(n_c)}$ |
| **Kim (2016) [87]** | – – – | $Historique\ du\ comportement\ des\ autres\ noeuds\ du\ meme\ cluster(au\ niveau\ du\ noeud\ CH)(format\ non\ spécifié)$ | $\geq \boldsymbol{O(n_c)}$ |

$n_c$ : nombre total de nœuds dans un cluster.

#### 4.3.3.2 Charge de communication

Dans [73], un seul message d'accusation envoyé à la CA conduit à la révocation de clés d'un nœud et celui-ci inclut quatre identités plus des informations supplémentaires, dont la taille est de 35 bits. La CA complétera alors le processus de révocation en envoyant aux différents nœuds du réseau les WL et BL mises à jour.

[87] par contre, ne donne aucun détail ni pour le contenu du message d'accusation envoyé par le nœud CH aux $n_c$ membres de son cluster, ni pour le contenu de votes envoyés par ces derniers en guise de réponse. Le protocole ne précise également pas le contenu du message de révocation, qui est envoyé par le nœud CH à tous les nœuds de son cluster pour terminer le processus.

**Tableau 4. 13 : Charge de communication des Protocoles Hybrides – Résumé de l'analyse**

| Protocoles Hybrides | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Lui et al. (2013) [73]** | *dépend de* $n_{a-CA}$ | *mess. accusation* | - 4 *IDs*<br>- *données compl.* | *unicast* | 1 | $4.IDs + WL + BL + c$ | $\boldsymbol{O(1)}$ |
| | *tout le réseau* | *mess. revocation* | - *warning list (WL)*<br>- *black list (BL)*<br>- *données compl.* | *broadcast* | 1 | | |
| **Kim (2016) [87]** | *cluster* | *mess. accusation* | – – – | *broadcast* | 1 | – – – | $\geq \boldsymbol{O(n_c)}$ |
| | | *vote* | – – – | *unicast* | $n_c$ | | |
| | $n_{CH-CA}$ | *transfert vote* | – – – | *unicast* | 1 | | |
| | *cluster* | *mess. révocation* | --- | *broadcast* | 1 | | |

$n_{a-CA}$ : nombre de sauts entre le nœud accusateur et la CA, $n_c$ : nombre de nœuds dans un cluster, $n_{CH-CA}$ : nombre de sauts entre le nœud CH et la CA.

#### 4.3.3.3 Charge de calcul

En ce qui concerne les protocoles hybrides, aucune opération cryptographique n'est spécifiée ni pour [73], ni pour [87]. En ce qui concerne les opérations liées au processus de gestion de la confiance, seul [87] présente pour le mécanisme de révocation de clé une charge supplémentaire. En

fait pour ce protocole, le poids de chaque nœud votant est évalué par le nœud CH avant de pouvoir comparer leur somme au quotient de révocation établi.

**Tableau 4. 14 : Charge de calcul des Protocoles Hybrides – Résumé de l'analyse**

| Protocoles Hybrides | Type nœuds | Activité | Nature Opération(s) | Nombre d'opérations / nœud | Charge / activité / nœud (complexité) | Nombre de nœuds participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| Lui et al. (2013) [73] | *nœud simple* | *mess. accusation* | $---$ | $---$ | $---$ | $---$ | $---$ | $---$ |
| | *nœud simple* | *vérif. mess accusation* | $---$ | $---$ | $---$ | $---$ | | |
| Kim (2016) [87] | *nœud CH* | *mess. accusation* | $---$ | $---$ | $---$ | $---$ | $---$ | $---$ |
| | *nœud simple* | *vote* | $---$ | $---$ | $---$ | $---$ | | |
| | *nœud CH* | *vérif. vote* | - $---$<br>- *calcul poids* | $n_p$ | $---$ | $---$ | | |
| | *nœud CH* | *mess. rév.* | $---$ | $---$ | $---$ | $---$ | | |
| | *nœud simple* | *vérif. mess rév.* | $---$ | $---$ | $---$ | $---$ | | |

#### 4.3.3.4 Délai ou latence de la révocation

Les protocoles hybrides proposés précisent juste le type de messages échangés lors du processus de révocation. [87] précise pour chaque type de message la portée de propagation ainsi que le mode de transfert employé. L'absence de description des opérations cryptographiques qui seront utilisées pour protéger ces échanges nous empêche d'estimer avec plus de précision le délai qui sera associé au mécanisme de révocation.

**Tableau 4. 15 : Délai de révocation des Protocoles Hybrides – Résumé de l'analyse**

| Protocoles Hybrides | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authen. | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|
| Lui et al. (2013) [73] | *mess. accusation* | *depend de* $n_{a-CA}$ | *broadcast* | $---$ | $n_{a-CA} . T_{transmit} + ---$ | $---$ |
| | *mess. révocation* | *tout le réseau* | $---$ | $---$ | $---$ | |
| Kim (2016) [87] | *mess. accusation* | *cluster* | *broadcast* | $---$ | $T_{transmit} + ---$ | $---$ |
| | *vote* | *cluster* | *unicast* | $---$ | $n'_c . T_{transmit} + ---$ | |
| | *transfert vote* | *depend de* $n_{CH-CA}$ | *unicast* | $---$ | $n_{CH-CA} . T_{transmit} + ---$ | |
| | *message rév.* | *cluster* | *broadcast* | $---$ | $T_{transmit} + ---$ | |

$n_{a-CA}$ : nombre de sauts entre le nœud accusateur et la CA, $n'_c$ : nombre maximum de sauts dans un cluster, $n_{CH-CA}$ : nombre de sauts entre le nœud CH et la CA, $T_{transmit}$: délai pour le transfert d'un bit de donnée d'un saut vers le suivant.

L'évaluation des protocoles hybrides résumée par les tableaux 4.12 – 4.15 ne peut être complétée car les auteurs ne fournissent pas tous les détails nous permettant de tirer des conclusions quant à leur efficacité.

## 4.4 Evaluation de l'Applicabilité des protocoles de révocation de clés dans les MANETs

### 4.4.1 Type de réseau

Les protocoles distribués et décentralisés étudiés sont applicables dans les MANETs auto-organisés puisque toutes les tâches liées au mécanisme de révocation de clés sont exécutées par les nœuds sans l'implication d'une autorité centralisée de confiance. Par contre, les protocoles hybrides qui dépendent d'un tiers de confiance ne sont pas appropriés au scénario des réseaux ad hoc auto-organisés.

### 4.4.2 Taille du réseau

L'établissement de clés pour sécuriser les communications entre les nœuds du réseau et ceux qui viennent de le rejoindre ne posent pas de problèmes majeurs, car les protocoles proposés pour les MANETs sont tous basés sur la cryptographie à clé publique. Le seul problème qui survient avec l'arrivée de nouveaux nœuds dans le réseau survient avec les protocoles distribués et ceux hybrides renforçant leur sécurité par un mécanisme de gestion de la confiance. En effet avec ces protocoles distribués, pour évaluer la fiabilité d'un nœud accusateur, les données renseignant sur l'historique du comportement de ce dernier sont requises. [82] ne précise ni le format de stockage de ces données, ni comment et quand elles seront transmises aux nouveaux nœuds rejoignant le réseau.

[77] suggèrent d'envoyer aux nouveaux nœuds (dès leur arrivée au réseau) les tables de profils comportant $n$ entrées afin qu'ils vérifient les correspondances. Approximativement, cela engendrera $n^2$ vérifications, ce qui peut être très couteux. Pour répondre à ce problème, [79] et [28] proposent d'envoyer les informations requises via l'envoi de listes de révocation de clés (données sous forme matricielle) après chaque accusation ou mise à jour. Ce qui permet aux nouveaux nœuds de prendre part au processus de révocation dès leur arrivée au réseau.

[83] ainsi que les protocoles hybrides n'abordent pas ce problème. Notons toutefois que les protocoles hybrides utilisant concept de cluster pourront stocker et échanger localement (au sein d'un cluster) les données relatives à l'historique des accusations. Ce qui permettra d'alléger le problème lié à l'arrivée de nouveaux nœuds dans le réseau.

En ce qui concerne le passage à l'échelle, les protocoles distribués sont de bons candidats puisqu'ils allègent :

*i)* la charge de calcul de chaque nœud ;

*ii)* la charge de communication et le délai soit en utilisant le transfert par broadcast des messages, soit en limitant la portée de propagation de ces derniers.

Cependant, lorsque nous considérons les charges de stockage, les protocoles tels que [77], [79] et [83] ne permettent plus le passage à l'échelle vue que la complexité associée à celle-ci est dépendante de la taille du réseau $n$.

En ce qui concerne les protocoles décentralisés, lorsque nous considérons uniquement la charge de communication, nous constatons que cette dernière reste très faible avec une complexité constante ($O(1)$) malgré l'accroissement du nombre de nœuds dans le réseau. Même si leur charge de communication est non dépendante de la taille du réseau $n$, il n'en est pas de même pour la charge de stockage pour les protocoles décentralisés. Ce qui freine leur possibilité de passage à l'échelle.

Pour les protocoles hybrides, le manque de détails nous empêche d'estimer avec plus de précisions les charges de stockage, les charges de communication, les charges de calcul et le délai de la révocation. Ce qui nous empêche de nous prononcer sur leur possibilité de passage à l'échelle. Nous pensons toutefois que l'utilisation du concept de cluster et l'implication de nœuds CH dans le processus de révocation de clés permettra de mieux appréhender la question.

### 4.4.3 Mobilité des nœuds

Nous nous intéressons à la mobilité du nœud selon qu'il soit le nœud censé participer au processus de révocation, ou qu'il soit le nœud ciblé pour la révocation.

Dans le cas où le nœud en déplacement est censé participer au mécanisme de révocation de clé, le problème majeur que pose la mobilité est lié à la perte de connectivité, perte qui engendre des problèmes de sécurité. Cependant, l'ensemble des protocoles de révocation proposés pour les MANETs se basent sur la cryptographie asymétrique ou sur la combinaison de celle-ci avec la cryptographie symétrique pour fournir les services d'authentification et d'intégrité, et ainsi protéger les échanges. Ce faisant, avec les protocoles utilisant la cryptographie asymétrique basée sur PKI, les nœuds n'ont pas besoin d'établir de nouveaux liens avec leurs nouveaux voisins lorsqu'ils se déplacent à travers le réseau. Par conséquent, lorsque le nœud se déplace, il pourra continuer à recevoir ou envoyer des messages d'accusation ou de révocation en toute sécurité sans engendrer un coût de stockage, de calcul ou de communication supplémentaire. Par contre avec les protocoles combinant la cryptographie symétrique et celle asymétrique (particulièrement les protocoles basés sur l'identité), suite à un déplacement vers un nouveau voisinage, le nœud aura besoin d'établir de nouveaux liens avec les nouveaux voisins par le calcul des clés de paire respectives, afin de pouvoir communiquer avec eux en toute sécurité ; d'où un coup supplémentaire associé à la mobilité.

Afin d'optimiser la consommation de ressources telles que la bande passante et l'énergie, et de gagner en termes de performance, certains protocoles distribués tels que [6], [28], [79] et [82] limitent la portée de diffusion ou de propagation des messages d'accusation. Dans ce cas, un nœud ciblé peut alors user de sa mobilité pour rejoindre un nouveau voisinage et ainsi échapper à la sanction et / ou faire échouer la procédure de révocation enclenchée contre lui. Pour contrer cette attaque d'itinérance, [6], [28] et [79] choisissent une portée de propagation en fonction de la vitesse de déplacement des nœuds, et de la durée de validité de la clé publique ou du ticket d'accès au réseau. De cette manière le nœud ciblé ne pourra atteindre un nouveau voisinage avant l'expiration de son ticket ou de sa clé publique. [82] par contre n'aborde point cette question et donc ne gère point la question liée à la mobilité des adversaires.

**Tableau 4. 16 : Applicabilité des Protocoles - Résumé de l'analyse**

| PROTOCOLES | | | Type de réseau | Taille du réseau | | Mobilité des nœuds |
|---|---|---|---|---|---|---|
| | | | | Simplicité d'ajout de nouveaux nœuds | Passage à l'échelle | |
| Distribués | Luo et al. (2004) [6] | | *auto organisés* | *Oui* | *Oui* | *Oui* |
| | Saxena et al. (2004) [8] | | *auto organisés* | *Oui* | *Oui* | *Oui* |
| | Zhang et al. (2006) [10] | | *auto organisés* | *Oui* | *Oui* | *Oui* |
| | Arboit et al. (2008) [77] | | *auto organisés* | *Non* | *Non* | *Oui* |
| | Fan et Gong (2008) [79] | | *auto organisés* | *Oui* | *Non* | *Oui* |
| | Hoeper et Gong (2009) [28] | | *auto organisés* | *Oui* | *Oui* | *Oui* |
| | Chaib et al. (2012) [82] | | *auto organisés* | *– – –* | *Oui* | *Non* |
| | Guo et al. (2014) [83] | | *auto organisés* | *– – –* | *Non* | *Oui* |
| Décentralisés | Moore et al. (2006, 2007) [84], [27] | crypto. asym. | *auto organisés* | *Oui* | *Non* | *Oui* |
| | | crypto. sym. | *auto organisés* | *Oui* | *Non* | *Oui* |
| Hybrides | Lui et al. (2013) [73] | | *Planifiés* | *– – –* | *Oui* | *Oui* |
| | Kim (2016) [87] | | *Planifiés* | *– – –* | *Oui* | *Oui* |

Pour résumer, nous donnons ci-dessous sous forme de tableau un aperçu sur les résultats de notre étude en listant pour chaque catégorie les avantages et inconvénients de chaque protocole.

**Tableau 4. 17 : Avantages et Inconvénients des Protocoles Distribués**

<table>
<tr><th>Protocoles Distribués</th><th colspan="2">AVANTAGES</th><th colspan="2">INCONVENIENTS</th></tr>
<tr><td>Luo et al. (2004) [6]</td><td>- limite la portée de propagation des messages (réduction charge de comm., délai et charge de calcul globale) ;</td><td rowspan="5">- Tous excepté Guo et al. (2014) [83] fournissent les services d'authentification et d'intégrité ;<br>- Propositions faites à partir de 2006 renforcent leur sécurité par un mécanisme de gestion de la confiance ;<br>- Appliquent des mesures visant à réduire les charges de communication (transfert par broadcast ou limitation de champs de propagation des messages) ;<br>- Conviennent bien aux MANETs ;<br>- Tous excepté Chaib et al. (2012) [82] gèrent la mobilité des nœuds.</td><td>- vulnérable à plusieurs types d'attaques (exhaus. batterie, fausse accusation, attaque par coalition, attaque sybil)</td><td rowspan="5">- premières propositions de solutions sont très vulnérables aux abus (le vote simple avec seuil ne suffit pas pour contrer ces attaques)</td></tr>
<tr><td>Saxena et al. (2004) [8]</td><td>- complète notification</td><td>- vulnérable à plusieurs types d'attaques (exhaus. batterie, fausse accusation, attaque par coalition, attaque sybil) ;<br>- charge globale de calcul élevée ($O(n)$)</td></tr>
<tr><td>Zhang et al. (2006) [10]</td><td>- complète notification</td><td>- vulnérable aux attaques par exhaustion de batterie, fausse accusation ;<br>- long délai de révocation (transfert de messages par unicast) ;<br>- charge globale de révocation élevée ($O(n)$)</td></tr>
<tr><td>Arboit et al. (2008) [77]</td><td>- complète notification ;<br>- n'utilise pas les signatures bien que basé sur PKI ;</td><td>- vulnérable aux attaques par exhaustion bat. (auth. du broadcast basée sur $\mu Tesla$) ;<br>- charge de stockage et charge globale de calcul importante ($O(n)$) (bloque le passage à l'échelle) ;<br>- ne gère pas simplement l'ajout de nœuds</td></tr>
<tr><td>Fan et Gong (2008) [79]</td><td>- résiste aux abus ;<br>- pas de vote (non vulnérable aux attaques par coalition) ;<br>- prend en compte les inexactitudes relatives au système de détection d'intrusion. (IDS)</td><td>- vulnérable aux attaques par exhaustion bat. (transfert par unicast de messages avec liste de révocation ayant $n$ entrées) ;<br>- charge de stockage importante ($O(n)$) ;<br>- long délai de révocation (transfert par unicast des</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | utilisé ; | | messages) | |

**Tableau 4. 17 : Avantages et Inconvénients des Protocoles Distribués (suite)**

| Protocoles Distribués | AVANTAGES | | INCONVENIENTS | |
|---|---|---|---|---|
| **Hoeper et Gong (2009) [28]** | - assure un bon niveau de sécurité (résiste aux abus) ;<br>- prend en compte les inexactitudes relatives au système de détection d'intrusion (IDS) utilisé ;<br>- facilite l'ajout de nœuds ; | | - possibilité d'accroissement de la charge de communication et du délai de révocation (transfert par unicast de messages) | |
| **Chaib et al. (2012) [82]** | - résiste aux abus ;<br>- pas de vote (non vulnérable aux attaques par coalition) ;<br>- plus faible charges de communication, de calcul globale et délai de révocation (transfert de messages uniquement vers le voisinage à un saut) | | - ne gère pas la mobilité des nœuds (vulnérabilité aux attaques d'itinérance) ;<br>- question liée à l'ajout de nœuds non abordée | |
| **Guo et al. (2014) [83]** | - complète notification | | - vulnérable aux attaques de type MIM (absence de service d'authentification et d'intégrité) ;<br>- charge de stockage et charge globale de calcul importante ($O(n)$) (bloque le passage à l'échelle) ;<br>- question liée à l'ajout de nœuds non abordée | |

**Tableau 4. 18 : Avantages et Inconvénients des Protocoles Décentralisés**

| Protocoles Décentralisés | | AVANTAGES | | INCONVENIENTS | |
|---|---|---|---|---|---|
| **Moore et al. (2006, 2007) [84], [27]** | **crypto. asym.** | court délai de révocation ($O(k)$, $k$ étant le nombre maximum de sauts à parcourir dans le réseau) | - assure une protection des messages ;<br>- non vulnérable aux abus (exhaustion bat., fausse accusation, attaque par coalition) ;<br>- faible charge de communication ;<br>- applicable aux MANETs (car auto-organisés) ;<br>- permet l'ajout de nœud | | - vulnérabilité aux attaques sybil, aux attaques par réplication et aux attaques par révocation ;<br>- charge de stockage et charges globales de calcul élevées (($O(n)$ ou $O(logn)$) (frein pour le passage à l'echelle) ; |
| | **crypto. sym.** | - réduit la charge de stockage (comparé à l'autre approche) | | - charge de comm. plus élevée ($O(logn)$) ;<br>- plus de calcul par nœud ($logn + 1$)opérations de hachage) (comparé à l'autre approche) ;<br>- délai de révocation plus long / l'autre approche ($O(k.logn)$) | |

**Tableau 4. 19 : Avantages et Inconvénients des Protocoles Hybrides**

| Protocoles Hybrides | AVANTAGES | | INCONVENIENTS | |
|---|---|---|---|---|
| **Liu et al. (2013) [73]** | - faible charge de communication | - Assurent un bon niveau de sécurité **si et seulement si les autorités hiérarchiques impliquées ne sont pas compromises.**<br>- Peuvent supporter le passage à l'échelle (grâce au concept de cluster introduit et à l'implication d'autorités hiérarchiques) | | - Les autorités impliquées (nœud CH ou CA) constituent des points de défaillance ;<br>- Ne donnent pas assez de détails permettant une meilleure évaluation ;<br>- Conviennent juste aux réseaux ad hoc planifiés (WSNs par exemple) |
| **Kim (2016) [87]** | | | - | |

## 4.5 Discussions et Conclusion

Dans ce chapitre, nous avons mené une étude évaluative et comparative des différentes catégories de solutions proposées pour la révocation de clés dans les MANETs. Les résultats obtenus montrent que :

- les premières propositions de la catégorie de protocoles distribués notamment [6], [8] et [10] présentent des vulnérabilités à plusieurs types d'attaques (attaque par exhaustion de batterie, attaque par fausse accusation, attaque par coalition de nœuds, attaque sybil). Pour apporter des solutions à ces problèmes, les propositions qui ont suivi, telles que [77], [79] et [82], évitent l'usage des signatures et /ou limitent la portée de propagation des accusations. Un mécanisme de gestion de la confiance est également utilisé pour combattre les éventuels abus. Ce qui améliore nettement le niveau de sécurité des propositions plus récentes (voir tableau 4.1) ;
- En ce qui concerne l'aspect efficacité, les protocoles distribués appliquent des mesures pour réduire les charges de communication et les charges individuelles de calcul en utilisant :
  - *i)* le broadcast ou en limitation de la portée de diffusion des messages ;
  - *ii)* le vote avec seuil (où chaque nœud vote une seule fois).

  Toutefois cela ne suffit pas pour les rendre aussi efficace puisque nombreux sont ceux parmi eux ([8], [10], [77], [79] et [83]), qui engendrent des charges de stockage et les charges globales de calcul dont la complexité associée est supérieure ou égale à $O(n)$. Ceci est principalement lié au stockage d'informations relatives à la vérification des accusations et à l'implication de l'ensemble des nœuds du réseau. Par conséquent ces protocoles soutiennent peu le passage à l'échelle et conviennent juste pour les réseaux de petite taille. Notons également que dû au transfert de messages par unicast, les protocoles distribués tels que [10], [79] et [28] engendrent une plus grande charge de communication et un délai de révocation assez long car ceux-ci dépendent d'autres paramètres tels que le nombre de sauts à traverser et le nombre de nœuds participants (voir tableau 4.5 et tableau 4.7) ;

- En ce qui concerne la mobilité, tous les protocoles distribués excepté [82], tiennent compte de cet aspect car soit :
  - *i)* utilisent la cryptographie à clé publique ;
  - *ii)* proposent la complète notification ;
  - *iii)* limitent la diffusion de l'information de révocation en choisissant le nombre de sauts en fonction de la validité des clés ou jetons des nœuds et leur vitesse de déplacement ;

- En éliminant le besoin de coopération entre les nœuds, les protocoles décentralisés proposent un moyen de contrer les abus. Cependant malgré cette mesure, la lourde sanction infligée au nœud accusateur (qui est le suicide), fait que les protocoles décentralisés ne sont pas à l'abri

de menaces telles que les attaques sybil, les attaques par réplication et les attaques par révocation.

- Comme certains protocoles distribués, les protocoles décentralisés préconisent la complète notification. Ce qui alourdit leur charge de stockage et leur charge globale de calcul, et empêche leur possibilité de passage à l'échelle ;
- En ce qui concerne leur applicabilité, les protocoles décentralisés proposant une approche distribuée conviennent aux réseaux ad hoc auto-organisés ;
- En combinant l'approche centralisée et celle distribuée, les protocoles hybrides héritent aussi bien des forces que des faiblesses de ces dernières. Une des forces de cette catégorie de protocole est leur possibilité de résister face à plusieurs types d'attaques, si les autorités hiérarchiques impliquées (nœuds CH et la CA) sont hors de portée des adversaires (voir tableau 4.3). Cependant, due à l'absence d'infrastructure fiable dans les MANETs [34], la satisfaction d'une telle condition ne peut être garantie. Ce qui remet en question la sécurité et l'applicabilité des protocoles hybrides dans l'environnement des MANETs.

Nous tenons toutefois à faire remarquer que bien que les deux protocoles hybrides étudiés ne soient pas explicitement conçus pour les WSNs, ils suggèrent de techniques et présentent des atouts qui font d'eux de potentiels candidats pour un déploiement dans de tels réseaux.

# Chapitre 5 ANALYSE DES PROTOCOLES DE RÉVOCATION DE CLÉS DANS LES WSNs

Les critères d'évaluation restent identiques à ceux décrits dans le chapitre précédent. Ainsi, nous nous intéressons en ce qui concerne l'aspect sécurité aux services fournis par les protocoles, aux attaques potentielles et à leur résistance face à ces dernières.
Pour ce qui est de l'aspect efficacité, nous évaluons les charges de stockage, les charges de communication, les charges de calcul et les délais ou latence de révocation des différents protocoles.
Concernant l'applicabilité des solutions proposées, nous déterminons le type de réseau dans lequel le protocole peut être mis en œuvre sans perdre ses propriétés en nous intéressant à la taille du réseau et à la mobilité de ses composants.

## 5.1 Evaluation de la sécurité des protocoles de révocation de clés dans les WSNs

### 5.1.1 Protocoles Centralisés

#### 5.1.1.1 Attaques de type Man-in-the-middle (MiM), Attaque DoS (exhaustion de batterie)

Pour fournir les services d'authenticité, d'intégrité et même quelques fois de confidentialité et garantir la protection des messages envoyés par l'autorité de confiance, les protocoles centralisés se basent principalement sur la cryptographie symétrique. Ce qui leur procure une protection contre les attaques de type *man-in-the-midlle*. Eschenauer et al. utilisent dans [91] la signature numérique pour assurer l'authentification de la diffusion. Cependant en cas de compromission de la clé de signature, le protocole peut faire l'objet d'attaques par révocation car un adversaire peut se faire passer pour le nœud contrôleur et diffuser de faux messages de révocation.

Les protocoles décrits dans [93] et [95], qui supposent l'utilisation d'une méthode d'authentification de la diffusion relative à µTesla (par exemple, [90]) ont la capacité de se défendre contre les attaques par révocation. Cependant cela les rend vulnérables aux attaques DoS, car cette méthode ne permet pas une authentification immédiate. En d'autres termes, les nœuds doivent transférer le paquet de diffusion reçu avant de l'authentifier correctement en raison de la divulgation retardée de la clé requise.

Dini et Savino fournissent dans [92] une méthode d'authentification de la diffusion utilisant une chaîne de hachage à sens unique. [29] et [96] assurent la fiabilité des échanges grâce aux clés de paire partagées entre la SB et les nœuds, celles-ci étant utilisées pour chiffrer les messages relatifs au processus de révocation.

#### 5.1.1.2 Attaque sybil, attaque par réplication de nœuds

La méthode de pré-distribution de clés basée sur le pool de clés rend [91] vulnérable aux attaques sybil. Avec ce protocole, un attaquant peut générer une ou plusieurs fausses identités, puisque plusieurs nœuds peuvent partager la même clé. [91] est également vulnérable aux « attaques via les clés de chemin additionnelles » décrites dans [117], puisqu'un nœud malveillant peut utiliser les clés sur son trousseau en tant que clés de chemin pour établir de nombreuses autres clés de chemin avec d'autres nœuds sans être détecté par la SB. De ce fait, même après la révocation des clés de ce dernier par la SB, les clés de chemin supplémentaires ne sont pas révoquées, et le nœud compromis en question pourra les utiliser pour rejoindre le réseau.

Pour prévenir les attaques sybil et garantir que seuls les nœuds valides existent dans le réseau, [93] et [95] utilisent des listes de révocation comportant les identités de tous les nœuds révoqués. Le certificat utilisé dans [96], empêche les nœuds malveillants de tenter d'utiliser plus d'une identité, prévenant ainsi les Attaques sybil. Par contre [29] se défait de toute utilisation de certificat à des fins d'authentification, mais le concept de cluster introduit lui permet d'adresser efficacement cette menace, puisqu'après l'arrivée d'un nouveau nœud dans son cluster, le nœud CH correspondant met à jour la liste des membres de son cluster et la valide auprès de la SB.

**Tableau 5. : Protocoles Centralisés – Résumé de l'analyse de sécurité**

| Protocoles Centralisés | Services fournis | | | Résistance aux potentielles attaques | | | | | | Niveau de Sécurité |
|---|---|---|---|---|---|---|---|---|---|---|
| | Authenticité | Intégrité | Confidentialité | MiM | Imper-sonation | Attaque Dos | Attaque par révocation | Sybil | Réplica-tion | |
| Eschnauer et Gligor (2002) [91] | ✓ | ✓ | × | ✓ | ✓ | ✓ | × | × | ✓ | Moyen |
| Dini et Savino (2006) [92] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Bon |
| Wang et al. (2007), Park et al. (2010) [93], [95] | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | ✓ | Convenable |
| Mansour et al. (2014) [96] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Bon |
| Seo et al. (2015) [29] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Bon |

✓ : le protocole dépend d'une hypothèse pour résister à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ = nombre d'attaques auxquelles le protocole résiste / nombre d'attaques potentilles considérées. Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x < 50\%$, Moyen : $50 \leq r_x < 75\%$, Convenable : $75 \leq r_x < 100\%$, Bon : $r_x = 100\%$.

Le tableau 5.1 montre qu'excepté le protocole Eschnauer et Gligor (2002) [91], tous les protocoles centralisés assurent au moins un niveau convenable de sécurité. Notons toutefois que cela est dû à l'implication d'une entité centralisée de confiance supposée hors de portée des adversaires.

#### 5.1.1.3 Unique Point de défaillance

D'après le tableau ci-dessus, tous les protocoles centralisés, excepté celui dans [91] assurent au moins un niveau convenable de sécurité, car parviennent à fournir les services requis et offrent une résistance face aux attaques potentielles. Cependant, il est à noter que toute la sécurité de ces protocoles repose entièrement sur celle de l'autorité centralisée, et pour tous les protocoles, cette dernière est supposée être digne de confiance, c'est-à-dire n'est pas susceptible d'être capturée. Cette hypothèse peut ne pas être vraie dans certaines applications, d'où l'existence pour ces protocoles d'un unique point de défaillance.

L'autorité centralisée doit en outre communiquer avec les nœuds afin d'obtenir des informations liées à d'éventuels comportements anormaux de nœud avant de pouvoir entreprendre des actions de révocation. Cela entraîne un long délai et donne à un adversaire plus de temps pour mener des attaques entraînant des conséquences plus dommageables.

### 5.1.2 Protocoles Distribués

Bien que les mécanismes de révocation distribués résolvent le problème associé à l'existence d'un unique point de défaillance inhérent aux protocoles centralisés, leur susceptibilité aux attaques contre le mécanisme de révocation est cependant très grande. En effet la mise en œuvre de la majorité de ces attaques ne demande que la compromission de nœuds participants alors que ces derniers bénéficient d'une faible protection physique.

#### 5.1.2.1 Attaques de type man-in-the-middle et attaques par exhaustion de batterie

Pour protéger les échanges, la quasi-totalité des protocoles distribués se basent sur la cryptographie symétrique pour fournir les services d'authentification, d'intégrité. Cette mesure, renforcée par la limitation de la portée de propagation des votes permet aux protocoles distribués de mieux combattre les attaques du genre *man-in-the-midlle* et les attaques par exhaustion de batterie. Le protocole présenté dans [102], combinant la cryptographie symétrique et la cryptographie asymétrique, assure la protection des votes, mais ne spécifie pas comment les messages chargés de diffuser la liste des nœuds révoqués seront authentifiés.

#### 5.1.2.2 Attaques par Déni de service (DoS)

Bien que plus résistants aux attaques par exhaustion de batterie, les protocoles distribués doivent faire face à d'autre formes d'attaques par déni de service. En effet, les adversaires peuvent supprimer les messages de révocation, voire s'accorder pour révoquer les nœuds non compromis. Sans une autorité centrale, la notification de chaque nœud de l'information de révocation devient une tâche difficile. Le protocole [98], par exemple, est vulnérable aux attaques DoS avec l'utilisation d'une diffusion à travers tout le réseau pour révoquer un nœud compromis, [27] souffre également de la même vulnérabilité due à la sollicitation de votes positifs pour la réélection d'un nœud légitime. Par contre, [18], [100] et [102] sont moins enclins aux attaques DoS car seuls les participants peuvent retransmettre les votes.

#### 5.1.2.3 Attaques sybil, attaques par réplication de nœuds

Dans [18], [98] et [100], les auteurs supposent juste que des techniques pour contrer les attaques sybil et les attaques par réplication de nœuds sont déjà en place. Cela constitue une faiblesse majeure car le problème du contrôle de ces attaques est un défi technique important lors de la conception d'une solution de révocation distribuée. Nous notons qu'avec l'utilisation de la clé de chemin, certains de ces schémas peuvent toutefois être vulnérables aux attaques sybil activées par les clés de chemin décrites dans [117]. Supposons qu'il existe un nœud malveillant $i$ qui a un lien direct avec le nœud $j$. $i$ peut générer plusieurs fausses identités et solliciter une clé de chemin pour chacune d'elles. Comme $j$ ne sait pas que ces membres votants sont de faux nœuds, $i$ peut utiliser les nœuds Sybil comme tampons de lancement.

Le protocole [102] est résistant aux attaques sybil car une vérification de certificat est requise lors de l'établissement de la communication. Le mécanisme proposé par [27] est également sécurisé contre les attaques sybil, car les nœuds peuvent se vérifier en reconstruisant et en présentant les jetons d'accès corrects pour chaque session.

#### 5.1.2.4 Attaques via les clés de chemin

La plupart des protocoles distribués [18], [98], et [100] se basent sur les clés symétriques pré-distribuées. Cependant, ces derniers ne parviennent pas à supprimer les clés de chemin associées à un nœud révoqué [117]. Cette faiblesse permet de nombreux autres types d'attaques susceptibles de compromettre le fonctionnement du protocole de révocation (par exemple un nœud malveillant peut rejoindre le réseau après sa révocation pour émettre des messages de révocation falsifiés et conserver l'accès aux clés de chemin établies pour les autres). Ainsi, afin d'assurer une bonne efficacité les mécanismes de révocation proposés, ces derniers doivent prendre en compte les clés de chemin dans les dernières étapes de la révocation des informations d'identification. Avec [102] ce problème ne se

pose pas car la cryptographie à clé publique basée sur PKI est utilisée. En exigeant la présentation d'un jeton valide à l'issue de chaque session, [27] résout également le problème.

#### 5.1.2.5 Attaques par fausse accusation, attaques par coalition de nœuds

Contrairement à la majorité des protocoles distribués étudiés avec les MANETs, ceux proposés pour les WSNs ne renforcent pas leur sécurité par le biais de mécanisme de gestion de la confiance. Les protocoles tels que [18], [98], [100] et [102] se basent uniquement sur le vote simple avec seuil, ce qui les rend vulnérables aux attaques par coalition de nœuds. Un nœud non compromis pourra alors facilement être révoqué par une coalition de nœuds malveillants dont le nombre atteint le seuil de révocation car ces derniers se contenteront de lancer chacun une attaque par fausse accusation. Dans [27], s'il existe de nombreux nœuds malveillants non détectés dans le réseau, ils peuvent supprimer des parts de jeton pour voter contre leurs voisins. Par conséquent, les nœuds non compromis ne seront pas en mesure de rejoindre le réseau en raison de membres votants insuffisants.

**Tableau 5. : Protocoles Distribués – Résumé de l'analyse de sécurité**

| Protocoles Distribués | Services fournis | | Résistance aux potentielles attaques | | | | | | | | Niveau de sécurité |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Authenticité | Intégrité | MiM | Exhaus. batterie | Dos | Sybil | Réplica-tion | Attaques via clés de chemin | Fausse accusation | Coalition de nœuds | |
| Chan et al. (2003) [98] | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | × | Faible |
| Chan et al. (2005) [18] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | Faible |
| Chao et al. (2013) [100] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | Faible |
| Chuang et al. (2010) [102] | × | × | × | ✓ | ✓ | ✓ | × | ✓ | × | × | Moyen |
| Moore et al. (2007) [27] | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | ✓ | ✓ | × | Moyen |

✓ : le protocole fournit le service ou résiste à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ = nombre d'attaques aux quelles le protocole résiste / nombre d'attaques potentilles considérées.
Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x < 50\%$, Moyen : $50 \leq r_x < 75\%$, Convenable : $75 \leq r_x < 100\%$, Bon : $r_x = 100\%$.

Les résultats donnés dans le tableau 5.2 ci-dessus montrent que les protocoles distribués proposés pour les WSNs sont sujets à plusieurs types d'attaques et ne parviennent pas à dresser une ligne de défense contre celles-ci, ce qui ne leur permet pas d'atteindre un niveau convenable de sécurité.

### 5.1.3 Protocoles Hybrides

Les protocoles hybrides garantissent une bonne protection des communications contre les attaques de type man-in-the-middle en fournissant les services d'authentification d'intégrité et même de confidentialité le plus souvent. Ils préviennent également les attaques par exhaustion de batterie en

limitant la portée de propagation des messages d'accusation. Tout comme les protocoles centralisés, ces protocoles se basent sur une ou plusieurs autorités centralisées chargées de compléter le processus de révocation en notifiant tout le réseau. Ces dernières sont supposées non susceptibles d'être compromises, ce qui leur procure une résistance face aux attaques par impersonation et aux attaques par révocation de nœuds légitimes.

Avec [103], une résistance aux attaques sybil et aux attaques par réplication de nœuds pourra également être fournie si l'enregistrement de l'emplacement de chaque nœud au sein du MSN est effectué et combiné à la vérification des identités de nœuds répliqués parmi les différents MSNs. Il est à noter qu'avec ce protocole aucune mesure n'est prise pour prévenir les attaques par fausse accusation et les attaques par coalition dont le but est de révoquer un nœud légitime.

Les mécanismes proposés par [106] proposent la suppression de toutes les clés de chemin construites avec ou à travers le nœud compromis et combinent cette mesure avec l'utilisation d'une liste de révocation (*BlackList*, BL). Ce qui leur garantit la révocation complète et une prévention contre les attaques sybil, les attaques par réplication de nœuds et la réinsertion de nœuds compromis dans le réseau. Malgré les améliorations apportées au protocole [18], [106] n'élimine pas le risque de d'abus via les nœuds qui lancent de fausses accusations. Ceci engendre une vulnérabilité face aux attaques par coalition de nœuds. Pour répondre à cette vulnérabilité, les auteurs se basent juste sur l'hypothèse de la non existence d'un nombre de nœuds compromis dans le voisinage d'un nœud supérieur ou égal au seuil de révocation $t_r$.

Le concept de cluster introduit par [107] et l'implication des nœuds CH dans la prise de décision de révocation élimine l'existence d'un unique point de défaillance et permet une protection contre les attaques par exhaustion de batterie, les attaques par fausse accusation et les attaques par coalition de nœuds. En fait avec ce protocole, au sein d'un cluster l'approche de révocation distribuée adoptée est similaire à [18]. Cependant le nœud CH est l'entité qui doit confirmer la compromission du nœud ciblé avant la diffusion de l'information. Ceci rend le mécanisme de révocation intra-cluster vulnérable aux attaques par révocation de nœud légitime car en cas de compromission de ce nœud CH, ce dernier pourra facilement révoquer des nœuds légitimes membres de son cluster. Cependant cette vulnérabilité est adressée car le protocole aborde également la question liée à la compromission du nœud CH.

La quasi-totalité des protocoles distribués basés sur la pré-distribution aléatoire de clés de paires rencontrent des vulnérabilités face aux attaques sybil, et aux attaques via les clés de chemin. [107] adresse efficacement ce problème en introduisant le concept de cluster associé au découpage de chaque cluster en sous-région et chaque pool en sous-pool.

[109] développé sur la base de [18] ne fournit aucune ligne de défense contre les attaques sybil, mais suppose juste que des fonctionnalités pour contrer ces dernières sont en place. L'implication de la SB pour compléter le processus en notifiant l'ensemble du réseau permet à ce protocole de lutter contre les attaques par réplication et les attaques via les clés de chemin. Due à l'utilisation de la méthode basée sur le vote simple basé sur un seul paramètre qui est le seuil de révocation, le système est également vulnérable aux attaques par fausse accusation et aux attaques par coalition de nœuds.

En combinant les méthodes centralisées et celles distribuées, les protocoles hybrides n'éliminent pas pour autant le risque de l'existence d'un unique point de défaillance. En effet comme avec les protocoles centralisés, pour ces protocoles, la décision de révocation revient finalement à l'autorité centralisée impliquée. Ce faisant, pour assurer leur sécurité, les protocoles hybrides se basent simplement sur l'hypothèse de l'existence d'autorités centralisées non susceptibles d'être capturées ou compromises.

**Tableau 5. : Protocoles Hybrides – Résumé de l'analyse de sécurité**

| Protocoles Hybrides | Services fournis | | Résistance aux potentielles attaques | | | | | | | | | Niveau de sécurité |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Authenticité | Intégrité | MiM | Exhaus. batterie | Imper-sonation | Attaque par Révocation | Sybil | Réplica-tion | Attaques clés de chemin | Fausse accusation | Coalition de nœuds | |
| **Sanchez et Baldus (2005) [103]** | ✓ | ✓ | ⓥ | ✓ | ⓥ | ⓥ | ✓ | ✓ | ✓ | × | × | **Convenable** |
| **Chattopadhyay et Turuk (2012) [106]** | ✓ | ✓ | ⓥ | ✓ | ⓥ | ⓥ | ✓ | ✓ | ✓ | × | ⓥ | **Convenable** |
| **Jiang et al. (2013) [107]** | ✓ | ✓ | ⓥ | ✓ | ⓥ | ⓥ | ✓ | ✓ | ✓ | ✓ | ✓ | **Bon** |
| **Ge et Choo (2014) [109]** | × | ✓ | ⓥ | ✓ | ⓥ | ⓥ | × | ✓ | ✓ | × | × | **Moyen** |

ⓥ : le protocole dépend d'une hypothèse pour résister à l'attaque, ✓ : le protocole fournit le service ou résiste à l'attaque, × : le protocole ne fournit pas le service ou ne résiste pas face à l'attaque. $r_x$ : taux de résistance du protocole $x$, avec $r_x$ = nombre d'attaques auxquelles le protocole résiste / nombre d'attaques potentilles considérées. Niveau de sécurité du protocole $x$ : Faible : $0 \leq r_x < 50\%$, Moyen : $50 \leq r_x < 75\%$, Convenable : $75 \leq r_x < 100\%$, Bon : $r_x = 100\%$.

Les résultats de l'analyse de sécurité listés dans le tableau 5.3 montrent que dans les WSNs, les protocoles hybrides parviennent à rehausser le niveau de sécurité des protocoles distribués en impliquant une autorité centralisée de confiance. Cependant tout comme dans les MANETs, dans les WSNs également l'assurance d'un niveau supérieur de sécurité dépend entièrement de la satisfaction d'une hypothèse : les autorités centralisées impliquées ne sont pas susceptibles d'être compromises.

## 5.2 Evaluation de l'Efficacité des protocoles de révocation de clés dans les WSNs

### 5.2.1 Protocoles Centralisés

#### 5.2.1.1 Charge de stockage

Pour [91] les nœuds ont juste besoin de stocker la clé personnelle partagée avec la SB, ainsi que la clé de signature obtenue après déchiffrement. Dans [92], chaque nœud stocke sa clé privée et les $h$ clés de son trousseau de clés, $h$ étant la hauteur de l'arbre de révocation. Si cet arbre est équilibré et est $a$-aire, $h$ sera alors egale à $log_a(n)$, $n$ étant le nombre total de nœuds dans le réseau. En ce qui concerne [93] et [95], aucun détail n'est précisé quant aux informations qui seront utilisées pour protéger les messages. Les auteurs mentionnent juste que les nœuds ont besoin de garder $s_{total}$ valeurs secrètes afin de pourvoir récupérer le contenu des messages de révocation lors des $s_{total}$ sessions. Le protocole [96] qui combine la cryptographie symétrique et celle asymétrique requiert pour chaque nœud le stockage de la clé publique de la SB, de sa propre paire de clés publique/privée et de la clé personnelle partagée avec la SB. Avec [29], qui se défait de toute utilisation de certificat pour participer au processus de révocation de clés, les nœuds ont juste besoin de stocker leur clé individuelle partagée avec la SB.

**Tableau 5. : Charge de stockage des Protocoles Centralisés – Résumé de l'analyse**

| Protocoles Centralisés | Données stockées par chaque nœud participant | Evaluation de la charge (Complexité) |
|---|---|---|
| **Eschnauer et Gligor (2002) [91]** | - 1 *clé personnelle*<br>- 1 *clé de signature* | $\boldsymbol{O(1)}$ |
| **Dini et Savino (2006) [92]** | - 1 *clé privée*<br>- $log_a n$ clés (trousseau de clés) | $\boldsymbol{O(log\ n)}$ |
| **Wang et al. (2007) et Park et al. (2010)** | - $s_{total}$ *valeurs secrètes*<br>- $---$ | $---$ |
| **Mansour et al. (2014) [96]** | - *la clé publique de la SB*<br>- 1 *paire de clés publique/privée*<br>- 1 *clé personnelle* | $\boldsymbol{O(1)}$ |
| **Seo et al. (2015) [29]** | - 1 *clé individuelle* | $\boldsymbol{O(1)}$ |

Le tableau 5.4 montrent que pour la majorité les protocoles centralisés, la charge de stockage produite au niveau de chaque nœud est faible puisque la complexité associée est égale à $O(1)$.

#### 5.2.1.2 Charge de communication

Avec [91], la clé de signature chiffrée est préalablement envoyée à chaque nœud du réseau par unicast, ensuite, pour révoquer les clés d'un nœud, le message de révocation est diffusé à travers le

réseau par broadcast et inclut tous les identificateurs de clés appartenant au trousseau de clés du nœud compromis, ainsi que la signature qui permettra la vérification du message. Dans [92], nous avons dans chaque message de renouvellement de clé envoyé par broadcast ou unicast la nouvelle clé chiffrée. Avec ce protocole, si l'arbre de révocation est $a$-aire, la diffusion de l'information de renouvellement de clés nécessitera alors $alog_a(n) - 1$ broadcasts et $(a-1)$ unicasts. Les protocoles [93] et [95] ne spécifient ni les informations à échangées, ni les calculs à effectuer afin de garantir la protection des communications lors du processus de révocation. Les auteurs mentionnent juste pour ces protocoles que chaque message de révocation est transmis par broadcast et inclut les identités des nœuds dont les clés sont à révoquer, et deux polynômes de faible degré.

Dans [29] et [96], les messages de révocation sont transmis par unicast aux différents nœuds du groupe ciblé. Avec [96] le groupe ciblé constitue l'ensemble des nœuds du réseau, et avec [29] le groupe cible est soit l'ensemble des nœuds CH du réseau (cas de compromission d'un nœud simple), soit l'ensemble des nœuds CH additionné à celui des nœuds du cluster, dont le nœud CH est visé pour la révocation (cas de compromission d'un nœud CH). Pour ce dernier protocole, nous obtenons dans les deux cas une charge globale dépendant du nombre de nœuds CH. Puisque ce nombre est strictement inférieur au nombre total de nœuds dans le réseau $n$, nous obtenons pour [29] une complexité inferieure à $O(n)$. Avec [96] le contenu de chaque message de révocation est la liste chiffrée contenant les identités des nœuds à révoquer. Par contre avec [29], un tel message est généré en chiffrant uniquement l'identité du nœud cible et une note donnant un aperçu sur le genre de nœud à révoquer (« *compNode* » ou « *compHeader* »).

**Tableau 5. : Charge de communication des Protocoles Centralisés - Résumé de l'analyse**

| Protocoles Centralisés | | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| **Eschnauer et Gligor (2002) [91]** | | *tout le réseau* | *message dist. clés* | *Clé de signature* | *unicast* | $n-1$ | $(n-1)$ (*clés de sign.*) $+1$ (*liste clés comp.*) | $\boldsymbol{O(n)}$ |
| | | | *message de révocation* | *Liste signée (clés compromises)* | *broadcast* | 1 | | |
| **Dini et Savino (2006) [92]** | | *tout le réseau* | *message renouv. Clés* | *prochaine clé chiffrée* | *unicast et broadcast* | $(a-1)$ *unicasts, et* $alog_a(n)-1$ *broadcasts* | $(a(1+log_a n) - 2)$(*clés chiffrées*) | $\boldsymbol{O(log\ n)}$ |
| **Wang et al. (2007), Park et al. (2010)** | | *tout le réseau* | *message de révocation* | *IDs nœuds comp.* $+$ *2 polynômes* | *broadcast* | 1 | --- | --- |
| | | *non spécifié* | *message d'auth.* | *non spécifié* | *non spécifié* | *non spécifié* | | |
| **Mansour et al. (2014) [96]** | | *tout le réseau* | *message de révocation.* | *liste chiffrée (IDs nœuds comp.)* | *unicast* | $n-1$ | $(n-1)$ (*listes chiffrées (noeuds comp.)*) | $\boldsymbol{O(n)}$ |
| **Seo et al. (2015) [29]** | **Rév. nœud simple** | *tout le réseau* | *message de révocation* | *Note chiffrée (ID + note de rév.)* | *unicast* | $n_{CH}$ | $n_{CH}$ *(Note chiffrée (ID + note de rév.))* | $< \boldsymbol{O(n)}$ |
| | **Rév. Nœud CH** | *tout le réseau* | *message de révocation* | *Note chiffrée (ID + note de rév.)* | *unicast* | $n_{CH} + n_{Cl}$ | $(n_{CH} + n_{Cl})$ (*Note chiffrée (ID + note de rév.)*) | |

$n$ : nombre total de nœuds dans le réseau, $n_{CH}$ : nombre total de nœuds CH dans le réseau, $n_{Cl}$ : nombre total de nœuds dans le cluster considéré

D'après le tableau 5.5 presque tous les protocoles centralisés génèrent une charge de communication qui dépend du nombre total d'éléments dans le réseau. En découpant le réseau en cluster pour ainsi ne notifier que les nœuds CH et éventuellement les nœuds du cluster concerné par la révocation, le protocole Seo et al. (2015) [29] parvient à réduire la charge de communication en la rendant indépendante du nombre total de nœuds dans le réseau.

#### 5.2.1.3 Charge de calcul

Dans le processus de révocation de clé, nous nous intéresserons juste aux opérations qui sont exécutées par les nœuds simples ou les nœuds CH et non à celles exécutées par la SB ou *sink,* puisque cette entité est considérée comme n'ayant pas de contrainte en termes de ressources de calcul.

Dans [91], chaque nœud du réseau doit exécuter une opération de déchiffrement afin d'obtenir la clé de signature, qui sera par la suite utilisée pour vérifier la signature du message de révocation. Dans [92], en cas de compromission de nœud, toutes les clés dans son trousseau deviennent compromises. Le KS (*Key Service*) doit alors renouveler ces dernières dans le trousseau des autres nœuds. Si $n_{cc}^{i}$ est le nombre de clés compromises dans le trousseau de clés du nœud $i$, ce dernier executera alors $n_{cc}^{i}$ déchiffrements et $n_{cc}^{i}$ opérations de hachage.

Avec [93] et [95], les opérations exécutées à des fins de vérification d'authenticité ne sont pas spécifiées. Pour ces deux protocoles, afin de reconstruire la clé de session, chaque nœud doit juste après réception du message de révocation évaluer deux polynômes à faible degré. Le protocole [96] requiert que chaque nœud dans le réseau exécute une opération de déchiffrement afin d'authentifier le message de révocation envoyé par la SB. Avec [29] en cas de compromission d'un nœud simple, le message révocation est envoyé aux différents nœuds CH du réseau, ce qui engendre un cout de calcul égal à $n_{CH}$ déchiffrements. En cas de compromission d'un nœud CH, le message de révocation sera envoyé aux différents nœuds CH du réseau, mais aussi aux nœuds du même cluster que le CH visé pour la révocation.

**Tableau 5. : Charge de calcul des Protocoles Centralisés - Résumé de l'analyse**

| Protocoles Centralisés | | Activité | Nature Opération | Nombre d'opérations / nœud | Charge / activité / nœud (complexité) | Nombre de nœuds participants / Rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| **Eschnauer et Gligor (2002) [91]** | | *récup. clé signature* | *déchiffrement* | 1 | $\boldsymbol{O(1)}$ | $n-1$ | $(n-1)$ (*déchif. et* (*vérif. sign.*) | $\boldsymbol{O(n)}$ |
| | | *vérif. mess. rév.* | *verif. signature* | 1 | | $n-1$ | | |
| **Dini et Savino (2006) [92]** | | *vérif. message de renouv.* | - *déchiffrement* - *hachage* | $n_{cc}^{i}$ | $\boldsymbol{O(1)}$ | $n-1$ | $\sum_{i=1}^{n-1} n_{cc}^{i}$ (*déchif. et hach*) | $\boldsymbol{O(n)}$ |
| **Wang et al. (2007) [93] et Park et al. (2010) [95]** | | *vérif. mess. rév.* | *non spécifié* | *non spécifié* | — — — | *non spécifié* | — — — | — — — |
| | | *reconst. secret* | - *calcul poly. univarié* (*faible degré*) | 2 | | $n-1$ | | |
| **Mansour et al. (2014) [96]** | | *vérif. mess. rév.* | *déchiffrement* | 1 | $\boldsymbol{O(1)}$ | $n-1$ | $(n-1)$ *déchiffrements* | $\boldsymbol{O(n)}$ |
| **Seo et al. (2015) [29]** | **Rév. nœud simple** | *vérif. mess. rév.* | *déchiffrement* | 1 | $\boldsymbol{O(1)}$ | $n_{CH}$ | $n_{CH}$ *déchiffrements* | $< \boldsymbol{O(n)}$ |
| | **Rév. nœud CH** | *vérif. mess. rév.* | *déchiffrement* | 1 | | $n_{CH} + n_{Cl}$ | $(n_{CH} + n_{Cl})$ *déchiffrements* | |

Le tableau 5.6 nous fait remarquer que pour les protocoles centralisés la charge de calcul générée par nœud est quasi négligeable puisque la complexité associée est de $O(1)$. Par contre pour tous les protocoles excepté Seo et al. (2015) [29], l'implication de tous les nœuds du réseau dans le processus de révocation des clés d'un seul nœud engendre une charge globale de révocation élevée dont la complexité est égale à $O(n)$.

#### 5.2.1.4 Délai ou latence de la révocation

Avec [91], les auteurs ne précisent pas à quel moment la clé de signature est envoyée aux nœuds du réseau. Nous supposons donc que cela se fait juste après détection d'un nœud compromis. Le nœud moniteur envoie alors par unicast à tous les nœuds du réseau la clé de signature chiffrée. Ce qui engendre un délai :

$$T_1 = n.T_{encrypt} + T_{transmit} \cdot \sum_{i=1}^{k} i.n_i + n.T_{decrypt} \qquad (5.1)$$

avec $n_i$ le nombre de nœuds situés à $i$ sauts par rapport a l'autorité centralisée, $1 \leq i \leq k$ et $k$ le nombre total de sauts du réseau. Pour terminer le processus de révocation de clés du nœud compromis le nœud moniteur doit envoyer par broadcast un message de révocation signé à l'ensemble des nœuds du réseau. Ce qui produit un délai :

$$T_2 = T_{sign} + k.T_{transmit} + n.T_{verif-sign} \qquad (5.2)$$

Nous obtenons alors pour ce protocole un délai total $T_{[91]}$ égal à :

$$T_{[91]} = T_1 + T_2.$$

$$T_{[91]} = n.[T_{encrypt} + T_{decrypt} + T_{verif-sign}] + T_{sign} + k.T_{transmit} + T_{transmit}.\sum_{i=1}^{k} i.n_i \quad (5.3)$$

De la même manière, nous évaluons le délai de révocation des autres protocoles centralisés basés sur le transfert par unicast de messages de révocation, notamment [29] et [96]. En utilisant le concept de cluster, [29] réduit le délai de révocation. En effet avec ce dernier protocole, le message de révocation n'est plus envoyé à tous les nœuds du réseau, mais uniquement aux nœuds CH du réseau dans le cas de compromission d'un nœud simple. Dans le cas de compromission d'un nœud CH, le message de révocation est envoyé à tous les nœuds CH du réseau, mais également à tous les nœuds simples du cluster concerné. Puisque les messages sont envoyés par unicast, au pire des cas, tous les nœuds simples du cluster concerné seront situés au dernier saut du réseau. Pour [29], la complexité associée au délai de révocation est inférieure à $O(n)$, étant donné que le nombre de nœuds CH présents dans le réseau additionné au nombre de nœuds du cluster concerné (cas de compromission d'un nœud CH) ne peut être que strictement inférieur au nombre total de nœuds $n$ du réseau.

En ce qui concerne [93] et [95], nous ne pouvons évaluer le délai de révocation, puisque le mécanisme d'authentification du broadcast n'est pas décrit. Pour réduire le délai de révocation, [92] diminue le nombre de messages de révocation et combine le transfert par unicast et par broadcast. Avec leur solution, lorsque l'arbre de révocation utilisé est équilibré et que la dimension de celui-ci est a, nous avons alors $(a-1)$ unicasts et $(alog_a n - 1)$ broadcasts. Ceci produit $(a(log_a n + 1) - 2)$ opérations de chiffrements et $\sum_{i=1}^{log_a n}(a^i - 1)$ opérations de déchiffrements et $\sum_{i=1}^{log_a n}(a^i - 1)$ opérations de hachage. Ceci engendre pour l'authentification des messages de renouvellement de clés un délai $T_1$ égal à :

$$T_1 = (a(log_a n + 1) - 2).T_{encryp} + \sum_{i=1}^{log_a n}(a^i - 1).T_{decryp} + \sum_{i=1}^{log_a n}(a^i - 1).T_{hash} \quad (5.4)$$

En ce qui concerne le transfert des messages de renouvellement de clés qui se fait soit par mode unicast soit par mode broadcast, nous évaluons le délai en considérant le pire des cas pour chaque mode de transfert. Ceci parce que nous ne connaissons pas la position des nœuds concernés.
Nous avons alors pour le mode de transfert par unicast, $a-1$ nœuds qui sont concernés. Au pire des cas, nous avons le maximum de nœuds parmi ceux qui sont concernés situés à $k$ sauts. Si $max_{a-1}$ est ce nombre, nous avons le délai :

$$T_{uni-max} = k.T_{transmit}.max_{a-1} \quad (5.5)$$

Pour le mode de transfert par broadcast, puisque $(alog_a n - 1)$ broadcasts sont requis, au pire des cas nous aurons pour chaque broadcast un nœud concerné situé au dernier saut. Ce qui donne au pire des cas un délai :

$$T_{bro-max} = k.T_{transmit} \,.\, (alog_a n - 1) \qquad (5.6)$$

Nous obtenons alors au pire des cas un délai de transfert :

$$\mathrm{T}_2 = T_{uni-max} + T_{bro-max} = \; k.T_{transmit} \,.\, (alog_a n - 1 + max_{a-1}) \qquad (5.7)$$

Le délai de révocation est alors obtenu en sommant le délai d'authentification et le délai de transfert déterminés ci-dessus.

**Tableau 5. : Délai de révocation des Protocoles Centralisés - Résumé de l'analyse**

| Protocoles Centralisés | | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authentification | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|---|
| Eschnauer et Gligor (2002) [91] | | *distribution clés de signature* | *tout le réseau* | *unicast* | *- chiffrement<br>- déchiffrement* | $n(T_{encryp} + T_{decryp}) + T_{transmit} \cdot \sum_{i=1}^{k} i.n_i$ | $\boldsymbol{O(k+n)}$ |
| | | *message de révocation* | *tout le réseau* | *broadcast* | *- signature<br>- verif.sign.* | $T_{sign} + n.T_{verif-sign} + k.T_{transmit}$ | |
| Dini et Savino (2006) [92] | | *message renouv. Clés* | *tout le réseau* | *- unicast<br>- broadcast* | *- chiffrement<br>- déchiffrement<br>- hachage* | $\leq (a(log_a n + 1) - 2).T_{encryp} + \sum_{i=1}^{log_a n} (a^i - 1).(T_{decryp} + T_{hash}) + k.T_{transmit} \,.\, (alog_a n - 1 + max_{a-1})$ | $\boldsymbol{O(k\,.log\,n)}$ |
| Wang et al. (2007), Park et al. (2010) | | *message de révocation* | *tout le réseau* | *broadcast* | – – – | – – – | – – – |
| Mansour et al. (2014) [96] | | *message de révocation* | *tout le réseau* | *unicast* | *- chiffrement<br>- déchiffrement* | $n(T_{encryp} + T_{decryp}) + T_{transmit} \cdot \sum_{i=1}^{k} i.n_i$ | $\boldsymbol{O(k+n)}$ |
| Seo et al. (2015) [29] | Rév. nœud simple | *message de révocation* | *tout le réseau* | *unicast* | *- chiffrement<br>- déchiffrement* | $n_{CH}(T_{encryp} + T_{decryp}) + T_{transmit} \cdot \sum_{i=1}^{k} i.n'_i$ | $< O(\boldsymbol{k+n})$ |
| | Rév. Nœud CH | *message révocation* | *tout le réseau* | *unicast* | *-chiffrement<br>- déchiffrement* | $\leq (n_{CH} + n_{Cl})(T_{encryp} + T_{decryp}) + T_{transmit} \cdot \sum_{i=1}^{k} i.n'_i + k.T_{transmit}.n_{Cl}$ | |

Le tableaux 5.7, nous fait remarquer que pour presque tous les protocoles centralisés, le délai de révocation dépend des deux paramètres qui affecte la taille du réseau (nombre total d'éléments $n$ et nombre total de sauts à parcourir $k$). Il nous permet également de noter que le protocole Seo et al. (2015) [29] parvient à réduire le délai en ne notifiant que les nœuds CH et éventuellement uniquement les nœuds du cluster concerné par la révocation.

### 5.2.2 Protocoles Distribués

Les protocoles distribués proposés pour les WSNs requièrent la connaissance à priori de la position de chaque nœud avant même son déploiement dans le réseau. Pour l'analyse de performance, nous admettons que le réseau se compose d'une SB, de $n$ nœuds et de $m$ nœuds votants ou participants pour chaque nœud. En ce qui concerne le protocole [27], nous considérons dans notre analyse de performance uniquement la première méthode décrite dans le chapitre 3 car la deuxième méthode donne juste une idée générale à propos de son fonctionnement.

#### 5.2.2.1 Charge de stockage

Dans [98], nous avons pour chaque nœud $m$ participants et chacun de ces nœuds participants est pré-chargé avec les $m$ clés de vote, $m.logm$ valeurs de hachage d'authentification, $m$ valeurs racines d'arbre de Merkle et $m$ secrets d'activation. [18], qui introduit le concept de session engendre un accroissement de l'espace de stockage requis par [98], car la taille qui était requise avec ce dernier est multipliée par $s_{total}$, qui est le nombre de sessions considéré. [100] réduit ce coût de stockage utilisant le concept de matrice de Blom [101]. Ainsi chaque nœud stocke $m.s_{total}$ votes chiffrés et $m.s_{total}$ fois $t_r$ éléments de la matrice publique. Dans [102] chaque nœud doit stocker $m$ valeurs de hachage pour chacun des $m$ nœuds votants, ceci uniquement dans le but de vérifier l'authenticité et l'intégrité des votes. Dans [27], chaque nœud est pré-chargé avec $m.s_{total}$ parts, $m.s_{total}$ autres parts hachées, une valeur racine d'authentification et $logn$ valeurs d'authentification de chemin.

**Tableau 5. : Charge de stockage des Protocoles Distribués – Résumé de l'analyse**

| Protocoles Distribués | Données stockées par chaque nœud participant | Evaluation de la charge (Complexité) |
|---|---|---|
| **Chan et al. (2003) [98]** | - $m$ *clés de vote*<br>- $m.(\log m + 1)$ *valeurs de hachage*<br>- $m$ *masques d'activation* | $\boldsymbol{O(m.\log m)}$ |
| **Chan et al. (2005) [18]** | - $m.s_{total}$ *clés de vote*<br>- $s_{total}.(1 + m + m.\log m)$ *valeurs de hachage*<br>- $m.s_{total}$ *masques d'activation* | $\boldsymbol{O(m.s_{total}.\log m)}$ |
| **Chao et al. ((2013) [100]** | - $m.s_{total}$ *votes*<br>- $m.s_{total}.t_r$ (*éléments de la matrice publique*) | $\boldsymbol{O(m.s_{total}.t_r)}$ |
| **Chuang et al. (2010) [102]** | - $m$ *valeurs de hachage*<br>- $---$ | $\geq \boldsymbol{O(m)}$ |
| **Moore et al. (2007) [27]** | - $m.s_{total}$ *parts de jeton*<br>- $m.s_{total}$ *valeurs de hachage (parts de jeton hachées)*<br>- $(1 + \log n)$ *valeurs de hachage* | $\boldsymbol{O(\log n)}$ |

Le tableau 5.8 montre qu'avec les protocoles distribués proposés pour les WSNs la charge de stockage au niveau de chaque nœud n'est plus négligeable comme c'est le cas avec les protocoles

centralisés, car dépend maintenant du nombre de participants $m$, et éventuellement d'autres paramètres tels que le seuil de révocation $t_r$, et le nombre de sessions envisagées $s_{total}$.

#### 5.2.2.2 Charge de communication

[98] adopte le vote avec seuil et chaque nœud vote en envoyant par broadcast son vote secret, accompagné des $logm$ valeurs d'authentification. Pour permettre la vérification de vote, chaque nœud doit au préalable envoyer à chacun de ses $m$ participants le masque d'activation. Avec [18] le processus est similaire à celui décrit pour [98]. Cependant avec ce dernier protocole, la valeur de hachage du polynôme de révocation obtenu doit être envoyée par broadcast à travers tout le réseau afin de compléter le processus de révocation.

Dans [100] également, chaque nœud doit recevoir de chacun de ses $m$ participants un masque d'activaton, qui permettra la vérification de votes reçus. Avec ce protocole qui utilise également le vote avec seuil, un message de vote contient le vote constitué de $t_r$ éléments d'une ligne de la matrice de vote et $t_r$ éléments d'une colonne de la matrice publique. Le processus de révocation sera complété une fois que le message de révocation, qui est la valeur de hachage de la ligne de partage secret sera diffusé à travers le réseau.

Dans [102] chaque vote reçu contient l'identité du nœud ciblé, la date de réception de la paire de clés publique/privée de la SB, l'identifiant de certificat du nœud compromis et une valeur de hachage unique. Un nœud compromis sera révoqué si au moins $t_r$ votes ou CRV (*Certificat Révocation vote*) contre ce dernier ont été collectés. Avec [27], pour être réélu, chaque nœud doit recevoir de ses membres votants au moins $t$ parts de jeton d'accès représentées chacune par une valeur de hachage. Avec ce protocole, seul le nœud ciblé pour la révocation ne va pas bénéficier d'une réélection. Nous obtenons alors pour tout le réseau une charge de communication égale à $t.(n-1)$ valeurs de hachage. Cependant pour terminer le processus de réélection, chacun des $n-1$ nœuds envoie par broadcast le jeton d'accès reconstitué, la valeur de hachage de fin de chaine et $logn$ valeurs d'authentification de chemin.

**Tableau 5. : Charge de communication des Protocoles Distribués - Résumé de l'analyse**

| Protocoles Distribués | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Chan et al. (2003) [98]** | *tout le réseau* | *envoi masque* | 1 *clé secrète* | *unicast* | $m$ | $m$ (*clés secretes*) + $t_r$ . (*log m +1*) *(valeurs de hach.)* | $\boldsymbol{O(m + t_r . log\ m)}$ |
| | | *vote* | (*log m +1*) *val. de hach.* | *broadcast* | $t_r$ | | |
| **Chan et al. (2005) [18]** | *tout le réseau* | *envoi masque* | 1 *clé secrète* | *unicast* | $m$ | $m$ (*clés secretes*) + *1 (valeurs de hach.)* | $\boldsymbol{O(m)}$ |
| | | *message* | 1 *val. de hach.* | *broadcast* | 1 | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | *rév.* | | | | | |
| | *voisinage* à 1 *saut* | *vote* | $(log\ m)$ *val. de hach.* | *broadcast* | $t_r$ | $t_r . log\ m$ *(valeurs de hach.)* | $\boldsymbol{O(t_r . log\ m)}$ |

**Tableau 5. 9 : Charge de communication des Protocoles Distribués - Résumé de l'analyse (suite)**

| Protocoles Distribués | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Chao et al. (2013) [100]** | *tout le réseau* | *envoi masque* | 1 *clé secrète* | *unicast* | $m$ | $m$ *(clés secretes)* + *1 (valeurs de hach.)* | $\boldsymbol{O(m)}$ |
| | | *message rév.* | 1 *val. de hach.* | *broadcast* | 1 | | |
| | *voisinage* à 1 *saut* | *vote* | $2vect.$ $2.t_r$ *éléments* | *broadcast* | $t_r$ | $2.t_r$ $(vect.(t_r$ *éléments))* | $\boldsymbol{O(t_r)}$ |
| **Chuang et al. (2010) [102]** | *voisinage* à 1 *saut* | *vote* | $ID, date, ID_{cert}$, *val. de hach.* | *broadcast* | $t_r$ | $t_r$ $(ID + date + ID_{cert} +$ *valeur de hach.*) | $\boldsymbol{O(t_r)}$ |
| | *tout le réseau* | *envoi liste rév.* | - 1 *liste de rév.* - – – – | *broadcast* – – – | 1 – – – | 1 (*liste de rév.*) – – – | $\geq \boldsymbol{O(1)}$ |
| **Moore et al. ( 2007) [27]** | *voisinage* à 1 *saut* | *vote* | 1 *val. de hach.* | *unicast* | $t\ .(n-1)$ | $(n-1)[\ t + log\ n +$ $1](valeurs\ de\ hach.) +$ $(n-1)$ *jetons* | $\boldsymbol{O(n\ .log\ n)}$ |
| | | *envoi jeton* | *jeton,* $(log\ n + 1)$ *val. de hach.* | *broadcast* | $n-1$ | | |

Le tableau 5.9 montrent que les tous protocoles distribués excepté Moore et al. (2007) [27] génèrent une charge de communication qui ne dépend plus de la taille du réseau $n$ mais du nombre de nœuds participants au processus de révocation $m$, et du seuil de révocation $t_r$. Il montre également que même si le protocole Moore et al. (2007) [27] a une charge de communication qui dépend du nombre total de nœuds dans le réseau $n$, la portée de propagation de ses messages ne se limite qu'au voisinage direct à un saut.

#### 5.2.2.3 Charge de calcul

Dans [98], après réception d'un vote, le nœud participant doit exécuter en premier une opération de hachage pour déchiffrer celui-ci. Ensuite pour vérifier son authenticité, il calcule et compare la valeur racine de l'arbre de Merkle, obtenue avec celle qu'il avait gardée en mémoire. Il exécute pour cela $logm + 1$ autres opérations de hachage. Puisque le vote avec seuil est appliqué pour ce protocole, nous obtenons alors pour chaque nœud participant une charge de calcul égale à $t_r(logm + 2)$ opérations de hachages. Dans [18] après déchiffrement du vote reçu, la vérification s'effectue avec les mêmes opérations que celles décrites avec [98]. Cependant pour ce protocole, pour compléter le processus de révocation, le nœud doit déterminer le polynôme de révocation en résolvant un système linéaire de degré $t_r$. Une fois celui-ci obtenu, le nœud doit en outre calculer la valeur de hachage de ce polynôme de révocation avant de l'envoyer par broadcast à travers tout le réseau. Avec ce protocole, comme avec [100], nous négligeons les opérations relatives à la détermination du polynôme de

révocation car ne sont exécutées que par un seul nœud. Chaque nœud participant non local recevant ce dernier message déterminera alors juste la valeur de hachage de celui-ci afin de vérifier son authenticité.

[100] exécute les mêmes opérations mentionnées ci-dessus pour le protocole [18] à l'exception du processus de vérification des votes reçus. Dans cette proposition, après vérification de l'authenticité d'un vote, un nœud doit calculer deux fois les produits d'un vecteur ligne de $t_r$ éléments et d'un vecteur colonne de $t_r$ éléments.

Dans [102], chaque nœud participant doit calculer au moins $t_r$ opérations XOR pour vérifier les $t_r$ votes requis par révocation. Pour compléter le processus de révocation de clés le nœud ayant reçu le nombre seuil de votes envoie aux autres participants non locaux une liste de révocation. Cependant, Chuang et al. ne précisent pas comment ce dernier message pourra être authentifié. [27] requiert, pour chaque vote ou part de jeton reçu, l'exécution d'une opération de hachage afin de la vérifier. Puisqu'au moins $t$ votes doivent être reçus avant de pouvoir reconstituer le jeton d'accès, ceci engendre un cout de calcul égal à $t$ opérations de hachage. Cependant pour ce protocole, la reconstitution du jeton elle-même nécessite l'exécution d'un système linéaire de degré $t$, $t$ etant le seuil de partage secret utilisé par ce protocole. Il est à noter que les autres protocoles distribués proposés pour les WSNs utilisent des votes négatifs et limitent ainsi le nombre de nœuds qui doivent prendre part au processus de révocation. A la différence de ces protocoles, [27] adopte l'approche de votes positifs et implique l'ensemble des nœuds du réseau, excepté le nœud qui est ciblé pour la révocation.

Afin de terminer le processus, chaque nœud ayant terminé la reconstitution de son jeton d'accès, doit transmettre ce dernier à ses voisins avec les valeurs d'authentification requises. Pour vérifier le message, chaque nœud voisin exécutera alors $\log n$ opérations de hachage afin de pouvoir comparer la valeur racine obtenue à celle stockée en mémoire.

**Tableau 5. 0 : Charge de calcul des Protocoles Distribués - Résumé de l'analyse**

| Protocoles Distribués | Activité | Nature Opération (s) | Nombre d'opérations / nœud | Charge / activité /nœud (complexité) | Nombre de nœuds participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Chan et al. (2003) [98]** | $vérif.vote$ | $hachage$ | $t_r\,.\,(log\,m+2)$ | $\boldsymbol{O(t_r\,.\,log\,m)}$ | $m$ | $t_r\,.\;m\,.(log\,m+2)$ $hachages$ | $\boldsymbol{O(t_r\,.\;m\,.log\,m)}$ |
| **Chan et al. (2005) [18]** | $vérif.vote$ | $-\,chiffrement$ $-\,hachage$ | - $t_r(chiffrement)$ - $t_r\,.(log\,m+2)$ $(hachage)$ | $\boldsymbol{O(t_r\,.log\,m)}$ | $m_l$ | $t_r\,.(m-m_{nl})\,.[chiff+$ $(log\,m+2)\,hach]$ | $\boldsymbol{O(t_r\;.\;m\;.log\,m)}$ |
| | $vérif.$ $mess.rév.$ | $hachage$ | 1 | $\boldsymbol{O(1)}$ | $m_{nl}+1$ | $(m-m_l+1)\,hach$ | $\boldsymbol{O(m)}$ |
| **Chao et al. (2013)** | $vérif.vote$ | $-\,chiffrement$ $-\,produit\,de$ $vect.$ | - $t_r(chiffrement)$ - $2.t_r$ $(prod.vect.lig.$ | $\boldsymbol{O(t_r)}$ | $m_l$ | $t_r\,.(m-m_{nl})\,.[chif.+$ 2 (prod.vect)] | $\boldsymbol{O(t_r\,.\;m)}$ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **[100]** | | | *et vect, col.*) | | | | |
| | *vérif. mess.rév.* | *hachage* | 1 | $\boldsymbol{O(1)}$ | $m_{nl}+1$ | $(m-m_l+1)$ *hach.* | $\boldsymbol{O(m)}$ |

**Tableau 5. 10 : Charge de calcul des Protocoles Distribués - Résumé de l'analyse (suite)**

| Protocoles Distribués | Activité | Nature Opération (s) | Nombre d'opérations / nœud | Charge / activité /nœud (complexité) | Nombre de nœuds participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|
| **Chuang et al. (2010) [102]** | *vérif. vote* | *Xor op. (ou exclusif)* | $t_r$ | $\boldsymbol{O(t_r)}$ | $m$ | $m\,.\,t_r$ *(XOR op.)* $+ - - -$ | $\geq \boldsymbol{O(t_r\,.m)}$ |
| | *vérif. mess.rév.* | $- - -$ | $- - -$ | $- - -$ | $- - -$ | | |
| **Moore et al. (2007) [27]** | *vérif. vote* | *hachage* | $t$ | $\boldsymbol{O(t)}$ | $n-1$ | $((n-1).t + logn.(n-1).\sigma)$ *hachages* | $\boldsymbol{O(n.logn)}$ |
| | *vérif. jeton* | *hachage* | $log\,n$ | $\boldsymbol{O(log\,n)}$ | $(n-1).\sigma$ | | |

Contrairement à l'approche centralisée, l'approche distribuée génère une charge de calcul non négligeable au niveau de chaque nœud participant. Le tableau 5.10 le confirme et nous fait noter que parmi les protocoles basés sur le vote négatif, le protocole Chao et al. (2013) [100] présente la plus petite charge de calcul par nœud (ne dépendant pas du nombre de participants) et la plus petite charge globale de calcul (complexités respectives associées étant $O(t_r)$ et $O(m\,.\,t_r)$ ). Il nous permet également de noter que protocole Moore et al. (2007) [27] utilisant une approche différente basée sur le vote positif présente une charge de calcul par nœud et une charge globale de calcul dépendant de la taille du réseau $n$ puisque les complexités respectives associées sont $O(log\,n)$ et $O(n\,.\,log\,n)$.

#### 5.2.2.4 Délai ou latence de la révocation

Avec les protocoles distribués, nous supposons qu'il existe au moins un des nœuds participants du nœud source du message de révocation, qui est situé à $k$ sauts par rapport à ce dernier, $k$ étant le nombre total de sauts à parcourir pour atteindre tous les nœuds du réseau. Pour les protocoles tels que [18], [27] et [100], nous négligeons le délai associé à la reconstitution du secret de révocation. Ceci parce que celui-ci est indépendant du nombre de nœuds $n$, dans le réseau, du nombre de participants $m$ et du nombre total de sauts $k$.

Dans [98], après détection d'un nœud compromis, le vote est envoyé par broadcast aux $m$ différents nœuds participants. Puisque chaque vote est authentifié après exécution de $log\,m+1$ opérations de hachage, et que pour qu'un nœud détecté soit révoqué il faut au moins $t_r$ votes, nous obtenons pour l'authentification des messages un délai $T_1$ égale à :

$$T_1 = t_r.k.(log\ m + 1).T_{hash} \qquad (5.8)$$

En ce qui concerne le transfert, nous avons un délai $T_2$ qui est de :

$$T_2 = t_r.k.T_{transmit} \qquad (5.9)$$

Nous obtenons alors pour Chan et al. (2003) [98] un délai de révocation $T$ égal à la somme des délais d'authentification et de transfert $T_1$ et $T_2$. La complexité associée à ce délai ne dépend pas du nombre de nœuds du réseau $n$, mais du seuil de révocation, du nombre de sauts $k$ et du nombre de participants $m$. Dans le cas où ces paramètres sont non variables, nous obtenons alors pour Chan et al. (2003) [98] un délai de révocation de complexité $O(1)$.

Avec les protocoles [98], et [100] les votes ne sont plus envoyés par broadcast à tous les $m$ participants à travers le réseau, mais juste aux nœuds participants voisins d'un saut. Ce qui produit pour le vote un délai $T_1$ égal à :

$$T_1 = t_r\ .T_{transmit} + t_r\ .(log\ m + 1).T_{hash} \qquad (5.10)$$

Le message de révocation envoyé pour compléter le processus produit un délai $T_2$ égal à :

$$T_2 = k\ .(\ T_{transmit}+T_{hash}) \qquad (5.11)$$

Avec Chuang et al. (2010) [102] le vote est également limité au voisinage à un saut. Ce qui engendre un délai $T_1$ égal à :

$$T_1 = t_r\ .(T_{XOR} + T_{transmit}) \qquad (5.12)$$

En ce qui concerne [27], si le processus de renouvellement de jeton commence en même temps pour tous les nœuds non compromis du réseau, le délai $T_1$ écoulé après réception de $t$ parts de jeton par chaque nœud est :

$$T_1 = t\ .(T_{transmit}+T_{hash}) \qquad (5.13)$$

Le délai $T_2$ lié à l'envoi du jeton ainsi obtenu aux voisins est de :

$$T_2 = (log\ n + 1)\ .T_{hash} + \ T_{transmit} \qquad (5.14)$$

**Tableau 5. 11 : Délai de révocation des Protocoles Distribués - Résumé de l'analyse**

| Protocoles Distribués | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authen. | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|
| **Chan et al. (2003) [98]** | *vote* | *tout le réseau* | *broadcast* | *hachage* | $t_r\ .k((logm + 1)T_{hash} + T_{transmit})$ | $\boldsymbol{O(t_r\ .\ k\ .log\ m)}$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Chan et al. (2005) [18]** | *vote* | *voisinage* | *broadcast* | $-$ *chiffrement* $-$ *hachage* | $t_r \cdot ((logm + 1)T_{hash} + T_{crypt} + T_{transmit})$ | $\boldsymbol{O(t_r \cdot logm)}$ |
| | *message rév.* | *tout le réseau* | *broadcast* | *hachage* | $k.(T_{hash} + T_{transmit})$ | $\boldsymbol{O(k)}$ |
| **Chao et al. (2013) [100]** | *vote* | *voisinage* | *broadcast* | $-$ *chiffrement* $-$ *prod. vect.* | $t_r \cdot (T_{prod-vect} + T_{crypt} + T_{transmit})$ | $\boldsymbol{O(t_r)}$ |
| | *message rév.* | *tout le réseau* | *broadcast* | *hachage* | $k.(T_{hash} + T_{transmit})$ | $\boldsymbol{O(k)}$ |
| **Chuang et al. (2010) [102]** | *vote* | *voisinage* | *broadcast* | *opération XOR* | $t_r \cdot (T_{XOR} + T_{transmit})$ | $\boldsymbol{O(t_r)}$ |
| | *message rév.* | *tout le réseau* | *broadcast* | $---$ | $---$ | $---$ |

**Tableau 5. 11 : Délai de révocation des Protocoles Distribués - Résumé de l'analyse (suite)**

| Protocoles Distribués | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authen. | Délai / type message | Délai de révocation (complexité) |
|---|---|---|---|---|---|---|
| **Moore et al. (2007) [27]** | *vote* | *voisinage* | *unicast* | *hachage* | $t \cdot (T_{hash} + T_{transmit})$ | $\boldsymbol{O(t)}$ |
| | *envoi jeton* | *voisinage* | *broadcast* | *hachage* | $(logn + 1) \cdot T_{hash} + T_{transmit}$ | $\boldsymbol{O(logn)}$ |

Les résultats listés dans le tableau 5.11 ci-dessus montrent qu'avec les protocoles distribués le délai de révocation est fonction de plusieurs paramètres. Les premières solutions présentent un délai dépendant des paramètres tels que seuil de révocation $t_r$ ou le seuil de partage secret $t$, le nombre de participants $m$ ou $n$ et éventuellement le nombre de total de sauts à parcourir. En améliorant la performance des propositions antérieures, Chao et al. (2013) [100] a fourni une solution qui rend le délai de révocation indépendant du nombre de nœuds participants au processus de révocation d'un autre nœud.

### 5.2.3 Protocoles Hybrides

Les auteurs dans [103] se contentent juste de mentionner que les nœuds doivent stocker les informations de révocation ou votes collectés localement afin de pouvoir les transmettre au MSB une fois que celle-ci se connecte au MSN. Aucune autre information n'est précisée pour spécifier les détails relatifs à ce protocole. De ce fait, en ce qui concerne l'analyse de performance pour les protocoles hybrides, nous nous intéressons principalement aux protocoles dans [106], [107] et [109].

#### 5.2.3.1 Charge de stockage

Chacune des deux propositions faites dans [106] requiert que chaque nœud stocke au plus quatre polynômes bi-variés de degré $7l'$, $l'$ étant le nombre maximum de nœuds dans une région hexagonale. Pour plus de robustesse face aux éventuels abus, chaque nœud maintient également deux listes notamment la liste de nœuds voisins considérés comme suspects et celle des nœuds compromis. Avec [107], au sein d'un cluster, chaque nœud doit stocker les identités de ses $m$ nœuds votants, $m$ votes, $m$

valeurs de hachage racine et $m\,.\,logm$ valeurs de hachage d'authentification de chemin. En ce qui concerne l'inter-cluster, chaque nœud CH doit stocker $log_a n_{CH}$ clés en plus de la clé privée partagée avec le KC (*Key center*), $a$ étant la dimension de l'arbre de gestion de clés. Notons que si le nombre de clusters est dépendant de la taille du réseau, nous obtenons pour l'inter-cluster une complexité de stockage égale à $O(n)$. Dans [109], chaque nœud stocke un polynôme. Après les phases d'établissement de clés, chaque nœud stocke les listes de paires de clés, la liste de révocation, les parts secrètes, le masque d'activation et les valeurs de hachage de ses $m$ membres votants.

**Tableau 5. 12 : Charge de stockage des Protocoles Hybrides – Résumé de l'analyse**

| Protocoles Hybrides | | Données stockées par chaque nœud participant | Evaluation de la charge (Complexité) |
|---|---|---|---|
| **Chattopadhyay et Turuk (2012) [106]** | | - 4 *polynômes bivariés*<br>- *liste des noeuds suspects*<br>- *liste des nœuds révoqués* | $\boldsymbol{O(1)}$ |
| **Jiang et al. (2013) [107]** | **Rév. nœud simple** | - $m$ *ID de nœuds votants*<br>- $m$ *votes*<br>- $m\,.\,(log\,m+1)$ *valeurs de hachage* | $\boldsymbol{O(m\,.\,log\,m)}$ |
| | **Rév. nœud CH** | - 1 *clé privée*<br>- $log_a n_{CH}$ *clés (clés du trousseau)* | $\leq \boldsymbol{O(log\,n)}$ |
| **Ge et Choo (2014) [109]** | | - 1 *polynôme univarié*<br>- $m$ *clés de paires*<br>- $m$ *parts secrètes*<br>- $m$ *masques d'activation*<br>- $m\,.\,(log\,m+1)$ *valeurs de hachage* | $\boldsymbol{O(m\,.\,log\,m)}$ |

### 5.2.3.2 Charge de communication

Dans la proposition I de [106] chaque nœud vote en envoyant à tous les nœuds voisins du nœud ciblé et à la SB un message incluant l'identité de ce dernier, accompagnée d'un polynôme uni-varié de degré $7k'$. Après avoir reçu $t_r$ votes, la SB complète la procédure de révocation en envoyant par broadcast un message de révocation contenant l'identité de nœud compromis, les clés compromises et $n_b$ polynômes uni-variés de degré $7k'$, $n_b$ étant le nombre de régions basiques. Dans la proposition II, pour voter contre un autre nœud, celui-ci doit envoyer au nœud moniteur de sa région un message contenant l'identité du nœud ciblé et un polynôme uni-varié de degré $7k'$. Après réception de $t_r$ votes authentifiés, le moniteur envoie alors à tous les nœuds de sa région un message de révocation incluant l'identité du nœud compromis, les clés compromises et au plus $k'$ valeurs d'autentification obtenues chacune à partir d'un polynôme tri-varié.

Dans [107] chaque nœud doit au préalable échanger son identité et sa clé de vote avec les autres nœuds qui sont ses membres votants. Au sein d'un cluster, le vote inclut la clé de vote du nœud votant

et $log_2 m$ valeurs de chachage. Pour révoquer les clés compromises, le nœud CH correspondant envoie par broadcast un message de révocation aux nœuds des sept sous-régions hexagonales relatives au nœud compromis. Ce dernier message comporte l'identité du nœud en question et la liste des clés de son trousseau. En ce qui concerne l'inter-cluster, si $a$ est la dimension de l'arbre de gestion de clés et $n_{CH}$ le nombre de nœuds CH dans le réseau, le KC aura besoin d'envoyer par broadcast $a.(log_{\mathrm{a}}(n_{CH}) - 1)$ messages de mise jour de clés, et par unicast $(a-1)$ autres messages de mise à jour de clés.

Dans [109], pour permettre la vérification de votes, durant la phase d'établissement de connexion, chaque nœud échange avec ses $m$ membres votants leur parts secrètes, masks et valeurs de hachage. Durant la phase de vote, le nœud votant envoie un message chiffré contenant son identité et la part secrète qu'il avait reçue du nœud ciblé. Après réception de $t$ votes authentifiés, le secret du nœud visé pour la révocation de ses clés est diffusé afin de permettre aux participants non locaux d'être informés à propos de la révocation de ce dernier. La SB se charge de compléter le processus en diffusant à travers tout le réseau un message de révocation. Cependant le contenu de ce dernier message n'est pas précisé par le protocole.

**Tableau 5. 13 : Charge de communication des Protocoles Hybrides - Résumé de l'analyse**

| Protocoles Hybrides | | Portée de propagation | Types de messages | Contenu des messages | Mode de transfert | Nombre de transferts / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|
| Chattopadhyay et Turuk (2012) [106] | Prop. I | *voisinage* à 1 *saut* | *vote* | $ID, poly.univ.$ | *broadcast* | $t_r$ | $t_r\ ID + t_r(poly.univ.)$ | $\boldsymbol{O(t_r)}$ |
| | | *tout le réseau* | message rév. | $-ID$ $-$ *liste clés comp.* $-n_b\ poly.univ.$ | *broadcast* | 1 | $1\ ID + n_b(poly\ univ.) + 1\ (liste\ clés\ comp.)$ | $\boldsymbol{O(1)}$ |
| | Prop. II | *région hexagonale* | *vote* | $ID, poly.univ.$ | *unicast* | $t_r$ | $t_r\ ID + t_r(poly.univ.)$ | $\boldsymbol{O(t_r)}$ |
| | | | *message rév.* | $-ID$ $-$ *liste clés comp.* $-k'val.authen.$ | *broadcast* | $n_r$ | $n_r\ ID + + n_r\ .k'\ val.d'authen. + n_r(listeclés\ comp.)$ | $\boldsymbol{O(1)}$ |
| Jiang et al. (2013) [107] | Rév. nœud simple | *cluster* | *établiss. connexion* | $-ID$ $-$ *clé de vote* | *unicast* | $m$ | $m\ ID + (m + t_r)$ (*cléés de vote*) $+ (t_r.log_{dim}m)$ (val.*hach.*) | $\boldsymbol{O(m)}$ |
| | | | *vote* | $-$ *clé de vote* $-\ log_{dim}m\ val.hac$ | *broadcast* | $t_r$ | | |
| | | *sept relatives sous − régions* | *message rév.* | $-ID$ $-$ *liste clés comp.* | *broadcast* | 1 | $1\ ID + 1$ (*liste clés comp.)* | $\boldsymbol{O(1)}$ |
| | Rév. nœud CH | *tout le réseau* | *message renouv. clés* | *prochaine clé* | *unicast et broadcast* | $(a-1)unicastseta.log - 1broadcasts$ | $(a.\ (log_a n_{CH} + 1) - 2\ clés$ | $\leq \boldsymbol{O(log\ n)}$ |
| Ge et Choo (2014) [109] | | *tout le réseau* | *établiss. connexion* | $-$ *part secrète,* $-$ *clé d'act.* $-val.de\ hach.$ | *unicast* | $t_r\ .m$ | $+1$ valeur secrète $+\ t_r\ .m$ (*parts secrète*) $+\ t_r\ .m$ (*clé d'act.*) $+ t_r\ .m$ (*val.de hach.*) $+ - - -$ | $\geq \boldsymbol{O(t_r.m)}$ |
| | | | *message rév.* | *valeur secrète* | *broadcast* | 1 | | |
| | | | *message rév.(SB)* | $- - -$ | $- - -$ | $- - -$ | | |
| | | *voisinage* à 1 *saut* | *vote* | $-ID$ $-$ *part secrète* | *broadcast* | $t_r$ | $t_r$ ( $ID +$ (*parts secrètes*)) | $\boldsymbol{O(t_r)}$ |

Les tableaux 5.12 et 5.13 nous permettent de noter que tous comme les protocoles distribués, la majorité des protocoles hybrides génèrent une charge de stockage et une charge de communication qui dépend du nombre de nœuds $m$ ou $n$ devant participer au processus de révocation de clé d'un autre nœud. Nous notons toutefois que le protocole Chattopadhyay et Turuk (2012) [106] est celui qui la charge de stockage et la charge de communication la plus faible car ne dépendant pas du nombre de participants (complexités respectives étant égale à $O(1)$ et $O(t_r)$).

#### 5.2.3.3 Charge de calcul

Dans la proposition I de [106], afin de vérifier un vote, un nœud a besoin d'évaluer un polynôme uni-varié de degré $7l'$. Ce calcul n'est pas requis avec la proposition II due à l'implication de nœuds moniteurs. Pour compléter la procédure de révocation, dans la proposition I un nœud doit vérifier l'authenticité du message de révocation envoyé par la SB en calculant un polynôme uni-varié de degré $7l'$ et un polynôme bi-varié de degré $7l' + 1$. Dans la deuxième proposition, un nœud aura juste besoin de déterminer son polynôme de vérification qui est un polynôme bi-varié de degré $7l' + 1$. Puisque $t_r$ votes sont requis pour la révocation du nœud compromis, nous obtenons alors pour la proposition I $(t_r + 1)$ évaluations de polynômes uni-variés et une évaluation de polynômes bi-variés et uniquement une évaluation de polynôme s bi-variés dans la proposition II.

Dans [107], la valeur racine doit être calculée au sein d'un cluster, afin de vérifier l'authenticité du vote après son déchiffrement. Pour ce faire, $log_2 m + 1$ valeurs de hachage doivent être calculées. Pour compléter la procédure de révocation, le nœud CH concerné envoie par broadcast la liste signée des clés compromises aux nœuds des sept sous-régions hexagonales relatives à celle du nœud concerné. Ces derniers pourront alors vérifier le message de révocation reçu en vérifiant la signature à l'aide de la clé publique du nœud CH source du message.
En ce qui concerne l'inter-cluster, chaque nœud $i$ CH non compromis recevra un nombre de messages de renouvellement de clé qui correspond au nombre de clés compromises dans son trousseau de clés. Si $n_{cc}^i$ est ce nombre nous auront alors pour chaque nœud $i$ CH $n_{cc}^i$ déchiffrements et $n_{cc}^i$ opérations de hachage.

Pour éviter la limitation due à la pré-distribution de vote rencontrée avec les autres protocoles distribués, [109] ajoute à la phase d'établissement de connexion une étape de génération de vote. Dans celle-ci, chaque nœud produit $m$ parts secrètes en évaluant $m$ fois le polynôme unique de degré $t_r - 1$ pre-chargé. Pour calculer le vote, chaque membre votant recevant au moins $t_r$ votes doit après $t_r$ chiffrements, résoudre un système linéaire de degré $t_r$ afin d'obtenir le secret. Chaque membre votant vérifiera alors le résultat en exécutant une opération de hachage afin de comparer la valeur obtenue

avec celle gardée en mémoire. Le nœud ayant obtenu le secret doit en outre exécuter deux opérations de chiffrement en utilisant pour l'une la clé secrète partagée avec la SB, et pour l'autre le masque qui avait été distribué par le nœud compromis. Après réception du message de révocation, les participants non locaux pourront alors vérifier ce dernier en effectuant une opération de déchiffrement et une opération de hachage. En ce qui concerne le message de révocation envoyé par la SB pour terminer le processus, [109] ne donne aucun détail relatif au mode de transfert de celui-ci ainsi qu'au mode d'authentification qui sera employé.

**Tableau 5. 14 : Charge de calcul des Protocoles Hybrides - Résumé de l'analyse**

| Protocoles Hybrides | | Type nœuds | Activité | Opération(s) | Nombre opérations / nœud / rév. | Charge / act. / nœud (complexité) | Nombre participants / rév. | Charge globale du système / rév. | Charge globale (Complexité) |
|---|---|---|---|---|---|---|---|---|---|
| Chattopadhyay et Turuk (2012) [106] | Prop . I | nœud simple | vérifier vote | calcul poly. univ. | $t_r$ | $\boldsymbol{O(t_r)}$ | $n^i$ | $(n^i . t_r + n) poly. univ + (n^i . t_r + n)\ poly. Biv.$ | $\boldsymbol{O(n)}$ |
| | | | vérifier message rev. | calcul poly. univ. et biv. | 1 | $\boldsymbol{O(1)}$ | $n$ | | |
| | Prop. II | nœud simple | vérifier message rév. | calcul poly. biv. | 1 | $\boldsymbol{O(1)}$ | $n$ | $npoly. Biv$ | $\boldsymbol{O(n)}$ |
| Jiang et al. (2013) [107] | Rév. nœud simple | nœud CH | génération message rév. | signature a clé pub. | 1 | $\boldsymbol{O(1)}$ | 1 | $1\ sign + m . t_r[déchif. + (log_2 m + 1)\ hachages] + n_{7sr}\ (vérif. sign.)$ | $\boldsymbol{O(m . t_r . log\ m)}$ |
| | | nœud simple | vérifier vote | − déchif. − hachage | $-t_r(déchif.) - t_r.(log_2 m + 1)(hach.)$ | $\boldsymbol{O(t_r . log\ m)}$ | $m$ | | |
| | | | vérifier message rév. | vérifier signature | 1 | $\boldsymbol{O(1)}$ | $n_{7sr}$ | | |
| | Rév. nœud CH | nœud CH | vérifier message renouv. | − déchif. − hachage | $n_{cc}^i$ (nb. clés comp. trouss.) | $\boldsymbol{O(1)}$ | $n_{CH} - 1$ | $\sum_{i=1}^{n_{CH}-1} n_{cc}^i\ (déchif. + hach$ | $\leq \boldsymbol{O(n)}$ |
| Ge et Choo (2014) [109] | | nœud simple | générer Vote | calcul poly univ. | $m$ | $\boldsymbol{O(m)}$ | $m$ | $m^2(poly\ univ.) + (t_r + 2)\ chiff + (m_l . t_r + m_{nl}) déchif + 1\ (résol. syst. lin.) + (1 + m_{nl})\ hachage + ---$ | $\geq \boldsymbol{O(m^2)}$ |
| | | | voter | chiffrement | 1 | $\boldsymbol{O(1)}$ | $t_r$ | | |
| | | | vérifier vote | déchifrement | $t_r$ | $\boldsymbol{O(t_r)}$ | $m_l$ | | |
| | | | reconst. secret | résolution syst. lin. | 1 | $\boldsymbol{O(t_r . t_r)}$ | 1 | | |
| | | | vérifier secret | hachage | 1 | $\boldsymbol{O(1)}$ | 1 | | |
| | | | gén. mess. rév. | chiffrement | 2 | $\boldsymbol{O(1)}$ | 1 | | |
| | | | vérifier message rév. | − déchif. − hachage | 1 | $\boldsymbol{O(1)}$ | $m_{nl}$ | | |
| | | | vérifier message rév. (SB) | − − − | − − − | − − − | − − − | | |

D'après le tableau 5.14 le protocole Chattopadhyay et Turuk (2012) [106] a la charge de calcul par nœud la plus faible puisque celle-ci dépend uniquement du seuil de révocation $t_r$. Cependant puisque ce protocole suggère la complète notification sa charge globale de calcul peut devenir importante avec l'accroissement de la taille du réseau.

#### 5.2.3.4 Délai ou latence de la révocation

Dans la proposition I de Chattopadhyay et Turuk (2012) [106], le vote est envoyé vers le voisinage à un saut par broadcast. Après réception de celui-ci, le nœud récepteur calcul un polynôme uni-varié pour pouvoir l'authentifier. Puisque $t_r$ réceptions de votes valides permettent la révocation de clés d'un nœud, nous obtenons pour le vote un délai $T_1$ égal à :

$$T_1 = t_r \,.\, (2.T_{poly} + T_{transmit}) \qquad (5.15)$$

Après réception de $t_r$ votes, le processus de révocation sera complété par la SB qui après calcul de $n_r$ polynômes bi-variés envoi par broadcast à tous les nœuds du réseau un message de révocation. A la réception de ce dernier, chaque nœud détermine un polynôme uni-varié et un polynôme bi-varié. En supposant que le délai pour déterminer un polynôme bi-varié est égal à $2\,.\,T_{poly}$, nous obtenons alors pour le message de révocation un délai $T_2$ égal à :

$$T_2 = 2.n_r\,.\,T_{poly} + k.\,(3\,.\,T_{poly} + .\,T_{transmit}) \qquad (5.16)$$

Avec la proposition II, le vote est transmis vers le nœud moniteur de la même région hexagonale. Puisque $t_r$ votes sont requis et que dans une région il y'a un nombre total de sauts égal à $k_r$, nous avons alors pour le vote un délai $T_1$ égal à :

$$T_1 = k_r.t_r\,.\,T_{transmit} \qquad (5.17)$$

Le message de révocation qui termine le processus produit un délai $T_2$ qui est de :

$$T_2 = k_r\,.\,(3\,.\,T_{poly} + T_{transmit}) \qquad (5.18)$$

puisque le nœud moniteur l'envoie par broadcast à tous les nœuds de sa région.

Dans [107], afin de révoquer un nœud simple, le vote est transmis par broadcast à tous les nœuds participants, et ces derniers se trouvent dans le même cluster. Pour vérifier un vote, le nœud récepteur aura besoin d'effectuer une opération de déchiffrement et une opération de hachage, d'où un délai $T_1$ égal à :

$$T_1 = t_r\,.\,\left(T_{decrypt} + T_{hash}\right) + k_c\,.\,T_{transmit} \qquad (5.19)$$

$k_c$ étant le nombre total de sauts à parcourir dans un cluster. Le message de révocation qui complète le mécanisme sera ensuite signé et envoyé par broadcast à tous les nœuds des sept régions hexagonales voisines. Ce qui génère un délai

$$T_2 = T_{sign} + 3.k_r\,.\,(T_{verif-sign} + .\,T_{transmit}) \qquad (5.20)$$

En cas de compromission d'un nœud CH, des messages de renouvellement de clés seront envoyés à travers le réseau pour révoquer les clés de ce dernier. Puisque le mécanisme de mise à jour de clés est similaire à celui décrit par [92], nous obtenons alors le même délai de révocation.

Avec [109], les votes sont limités au voisinage à un saut et ces derniers sont authentifiés par le biais de mécanismes de chiffrement et déchiffrement. Le délai associé est alors :

$$T_1 = T_{crypt} + t_r.\,(T_{decrypt} + T_{transmit}) \qquad (5.21)$$

Le nœud ayant accumulé les $t_r$votes poursuit le processus en envoyant par broadcast un message de révocation chiffré à tous les participants non local et à la SB. Ce qui engendre un délai :

$$T_2 = 2.T_{crypt} + (k-1)\,.\,(T_{decrypt} + T_{hash} + T_{transmit}) \qquad (5.22)$$

Pour permettre une complète notification, les auteurs de [109] proposent de compléter le processus en impliquant la SB pour l'émission de messages de révocation qui doivent propager à travers tout le réseau. Cependant, Ge et Choo ne donnent aucun détail quant au mode de transfert de ce dernier message ni à son authentification.

**Tableau 5. 5 : Délai de révocation des Protocoles Hybrides - Résumé de l'analyse**

| Protocoles Hybrides | | Types de messages | Portée de propagation | Mode de transfert | Opérations d'authentification | Délai / type message | Délai de révocation (Complexité) |
|---|---|---|---|---|---|---|---|
| Chattopad-hyay et Turuk (2012) [106] | Prop . I | *vote* | *voisinage* | *broadcast* | *calcul poly.* | $t_r\,.\,(2.T_{poly} + T_{transmit})$ | $\boldsymbol{O(t_r)}$ |
| | | *message rév.* | *tout le réseau* | *broadcast* | *calcul poly.* | $2.n_r\,.T_{poly} + k.(3\,.T_{poly} + .T_{transmit})$ | $\boldsymbol{O(k)}$ |
| | Prop. II | *vote* | *région hexag.* | *unicast* | | $k_r.t_r\,.T_{transmit}$ | $\boldsymbol{O(t_r)}$ |
| | | *message rév.* | *région hexag.* | *broadcast* | *calcul poly.* | $k_r\,.(3\,.T_{poly} + T_{transmit})$ | $\boldsymbol{O(1)}$ |
| Jiang et al. (2013) [107] | Rév . nœud simple | *vote* | *cluster* | *broadcast* | $-$ *déchiffrement.* $-$ *hachage* | $t_r\,.(T_{decrypt} + T_{hash}) + k_c\,.T_{transmit}$ | $\boldsymbol{O(t_r)}$ |
| | | *message rév.* | *7 relatives sous − régions* | *broadcast* | $-$ *signature* $-$ *vérif.sign.* | $T_{sign} + 3.k_r\,.(T_{verif-sign} + .T_{transmit})$ | $\boldsymbol{O(1)}$ |
| | Rév . nœud CH | *message renouv. clés* | *tout le réseau* | *unicast et broadcast* | $-$ *chiffrement* $-$ *déchiffrement* $-$ *hachage* | $\leq (a(log_a n + 1) - 2).T_{encryp} + \sum_{i=1}^{log_a n}(a^i - 1)\,.(T_{decryp} + T_{hash}) + k.T_{transmit}\cdot(a log_a n - 1 + max_{a-1})$ | $\boldsymbol{O(k.log\ n)}$ |
| Ge et Choo (2014) [109] | | *vote* | *voisinage* | *broadcast* | $-$ *chiffrement* $-$ *déchiffrement* | $T_{crypt} + t_r.\,(T_{decrypt} + T_{transmit})$ | $\boldsymbol{O(t_r)}$ |
| | | *message rév.* | *tout le réseau* | *broadcast* | $-$ *chiffrement* $-$ *déchiffrement* $-$ *hachage* | $2.T_{crypt} + (k-1)\,.\,(T_{decrypt} + T_{hash} + T_{transmit})$ | $\boldsymbol{O(k)}$ |
| | | *message rév. (SB)* | *tout le réseau* | $---$ | $---$ | $---$ | $---$ |

Bien que propageant les messages dans tout le réseau, Chattopadhyay et Turuk (2012) [106] produit un délai de révocation qui dépend de deux paramètres notamment le seuil de révocation $t_r$ et le nombre total de sauts à parcourir $k$. Par contre Jiang et al. (2013) [107] engendre un délai de révocation qui dépend du nombre total de sauts $k$, mais également du nombre d'éléments dans le réseau $n$.

## 5.3 Evaluation de l'Applicabilité des protocoles de révocation de clés dans les WSNs

Puisque les WSNs représentent un type de réseau ad hoc planifié, pour ce critère, nous nous intéressons alors juste à la taille du réseau et à l'aspect mobilité des nœuds.

### 5.3.1 Taille de réseau

Malgré leur possibilité de gérer de façon simple l'ajout de nouveaux nœuds, les protocoles centralisés n'offrent pas pour autant la possibilité pour le passage à l'échelle. Ceci est due au fait que pour ces derniers, les charges de communication, les charges globales de calcul ainsi que le délai de révocation sont tous dépendants de la taille du réseau $n$ (nombre de nœuds dans le réseau).

Pour adresser la question relative au passage à l'échelle, les protocoles distribués et la quasi-totalité des protocoles hybrides proposent les techniques basées sur le vote avec seuil et /ou le transfert local et/ou le transfert par broadcast. Ce faisant, pour ces derniers, les charges de communication, charges globales de calcul et le délai de révocation deviennent indépendants de la taille du réseau. Cependant il faut noter que malgré ces améliorations, les protocoles distribués et le protocole [107] requièrent la connaissance à priori de la position des nœuds après le déploiement, et limitent chaque nœud à un ensemble de voisins votants. Ce qui constitue un frein pour leur passage à l'échelle car les nouveaux nœuds ne pourront pas rejoindre facilement le réseau et participer à d'éventuels processus de révocation de nœuds voisins compromis.

### 5.3.2 Mobilité des nœuds

La naissance d'un nouveau type de WSN appelé réseau de capteurs dynamique fait de la gestion de la mobilité dans les WSNs un thème assez récent, contrairement aux MANETs. Ceci étant, pour pouvoir servir dans les WSNs dynamiques, un mécanisme de révocation de clés doit être en mesure de s'exécuter efficacement en dépit du mouvement des nœuds en leur permettant de communiquer de façon sécurisée et efficace avec leurs nouveaux voisins.

Pour presque la totalité des protocoles centralisés, la mobilité ne pose pas de problème de sécurité et/ou de performance, vu que les entités qui doivent prendre part au processus de révocation, notamment l'autorité centralisée et les différents nœuds du réseau, restent connectés malgré leur déplacement. De plus pour ces protocoles centralisés, les messages de révocation sont diffusés à travers tout le réseau, ce qui décourage les adversaires qui tenteraient de faire échouer le mécanisme de révocation en usant de leur mobilité (attaque par itinérance). Le protocole [92] qui se base sur la technique de la hiérarchie des clés logiques (*Logical Key Hierarchy (LKH)*) pour mettre à jour les clés des nœuds non compromis doit par contre être déployé pour les réseaux de capteurs statiques (avec des nœuds immobiles). Ceci parce que cette dernière technique établit ou définit un arbre logique en se basant sur la position physique des nœuds.

Les protocoles distribués et la quasi-totalité des protocoles hybrides proposés sont conçus uniquement pour un sous-ensemble d'applications dans lesquelles les nœuds sont supposés être immobiles. Ceci parce que la mobilité crée des complications que les protocoles ne parviennent pas à prendre en charge. En effet la première complication est liée au fait que pour ces protocoles seuls les voisins immédiats d'un nœud lors de la phase d'initialisation sont habilités à voter contre ou en faveur de ce nœud. Ce qui représente un obstacle pour la mobilité car cela retire aux nœuds la possibilité d'interagir avec de nouveaux voisins suite à leur déplacement. Une autre complication survient avec ces protocoles distribués basés sur le vote négatif ([18], [98], [100]), qui limitent la portée de diffusion des informations de révocation sans parvenir à supprimer les clés de chemin associées à un nœud révoqué. Ce qui donne à un nœud malveillant la possibilité d'user de sa mobilité pour changer d'emplacement et continuer à participer au réseau.

**Tableau 5. 16 : Applicabilité des Protocoles - Résumé de l'analyse**

| | PROTOCOLES | Taille du réseau | | Mobilité des nœuds |
|---|---|---|---|---|
| | | Ajout de nœuds | Passage à l'échelle | |
| Centralisés | **Eschnauer et Gligor (2002) [91]** | *Simple* | *Non* | *Oui* |
| | **Dini et Savino (2006) [92]** | *Simple* | *Non* | *Non* |
| | **Wang et al. (2007), Park et al. (2010) [19], [95]** | *Simple* | *Non* | *Oui* |
| | **Mansour et al. (2014) [96]** | *Simple* | *Non* | *Oui* |
| | **Seo et al. (2015) [29]** | *Simple* | *Non* | *Oui* |
| | | | | |
| Distribués | **Chan et al. (2003) [98]** | *Pas simple* | *Non* | *Non* |
| | **Chan et al. (2005) [18]** | *Pas simple* | *Non* | *Non* |
| | **Chao et al. (2013) [100]** | *Pas simple* | *Non* | *Non* |
| | **Chuang et al. (2010) [102]** | *Pas simple* | *Non* | *Non* |

| | | | | |
|---|---|---|---|---|
| | **Moore et al. (2007) [27]** | *Pas simple* | *Non* | *Non* |
| | | | | |
| **Hybrides** | **Sanchez et Baldus (2005) [103]** | *Simple* | *– – –* | *Oui* |
| | **Chattopadhyay et Turuk (2012) [106]** | *Simple* | *Oui* | *Non* |
| | **Jiang et al. (2013) [107]** | *Pas simple* | *Non* | *Non* |
| | **Ge et Choo (2014) [109]** | *Simple* | *– – –* | *Non* |

Ci-dessous, nous donnons, sous forme de tableaux un aperçu sur les résultats de notre étude en listant les différents avantages et inconvénients des différentes catégories de protocoles de révocation de clés étudiées pour les WSNs.

**Tableau 5. 17 : Avantages et Inconvénients des Protocoles Centralisés**

| **Protocoles Centralisés** | **AVANTAGES** | | **INCONVENIENTS** | |
|---|---|---|---|---|
| **Eschnauer et Gligor (2002) [91]** | | - Presque tous présentent au moins un niveau convenable de sécurité **(si et seulement si l'autorité centralisée est hors de portée des adversaires)** ;<br>- moins complexes que les autres types de protocoles (moins de vulnérabilités face à la compromission de nœuds ;<br>- faibles charges de stockage et de calcul ;<br>- tous excepté Dini et Savino (2006) [92] n'ont pas de contraintes empêchant la mobilité des nœuds ou de la SB.<br>- | - plus faible niveau de sécurité (utilisation d'une clé unique de signature) | - se basent tous sur une hypothèse simplificatrice: supposent que l'autorité centralisée est hors de portée des adversaires.<br>- existence d'un unique point de défaillance (la sécurité des protocoles dépend entièrement de celle de l'autorité centralisée) ;<br>- problème pour authentifier le broadcast, d'où le recours a l'unicast ;<br>- charge de communication et charge globale de calcul élevées (absence de possibilité d'authentification du broadcast et usage du mode unicast pour notifier de façon sécurisée tous les nœuds du réseau) ;<br>- long délai de révocation (le plus souvent dû à la notification de tous les nœuds du réseau via le transfert de l'information par unicast) ;<br>- ne supportent pas le passage à l'échelle. |
| **Dini et Savino (2006) [92]** | - réduit la charge de communication et le délai de révocation (complexité = $O(log\ n)$) | | - ne permet pas la mobilité (technologie basée sur les LKH) | |
| **Wang et al. (2007), Park et al. (2010) [19], [95]** | - possibilité de réajuster la durée d'une session avec Park et al. (2010) ;<br>- l'usage du transfert par broadcast permet d'écourter le délai de diffusion de l'information de révocation ;<br>- propose la révocation simultanée de plusieurs nœuds compromis | | - aucun détail sur l'authentification du message de révocation envoyé par boadcast | |
| **Mansour et al. (2014) [96]** | - propose la révocation simultanée de plusieurs nœuds compromis | | - le mécanisme gestion de clés sous-jacent est basé sur la crypto. asym. , mais pas celui de la révocation | |
| **Seo et al. (2015) [29]** | - introduit le concept de cluster pour plus de sécurité et de performance ; | | - le mécanisme gestion de clés sous-jacent est basé sur la crypto. asymétrique, mais pas celui de la | |

| | | | | |
|---|---|---|---|---|
| | - parvient à réduire la charge de communication. | | révocation | |

**Tableau 5. 18 : Avantages et Inconvénients des Protocoles Distribués**

| Protocoles Distribués | AVANTAGES | | INCONVENIENTS | |
|---|---|---|---|---|
| **Chan et al. (2003) [98]** | | - Eliminent l'existence d'un unique point de défaillance ;<br>- Sont presque tous résistants aux attaques de type MiM et attaques par exhaustion de batterie (limitation de la portée de propagation des accusations) ;<br>- tous excepté Moore (2007) sont plus performant que les protocoles centralisés (complexité de la charge de communication, charge globale de calcul, délai de révocation non dépendante de la taille du réseau). | - Vulnérable à plusieurs types d'attaques;<br>- Plusieurs failles de sécurité liées au mécanisme de pré-distribution de clés sous – jacent. | - Offrent au meilleur des cas un niveau moyen de sécurité ;<br>- Absence de mécanisme de gestion de la confiance pour contrer les abus (tous excepté Moore (2007) qui utilise le vote positif sont vulnérables aux attaques par fausse accusation et aux attaques par coalition de nœuds) ;<br>- ne fournissent pas une complète notification (vulnérabilités aux attaques sybil, attaques par réplication de nœuds, attaques via les clés de chemin, …) ;<br>- plus complexes que les protocoles centralisés (potentiels participants compromis) ;<br>- moins performants que les protocoles centralisés en termes de d'usage de ressources de stockage et de calcul au niveau de chaque nœud participant;<br>- ne supportent ni l'ajout de nouveaux nœuds, ni la mobilité des |
| **Chan et al. (2005) [18]** | - Adresse certaines insuffisances de Chan et al. (2003) [98] ; | | - Vulnérable à plusieurs types d'attaques ;<br>- le concept de session alourdit la charge de stockage | |
| **Chao et al. (2013) [100]** | - Plus performant que Chan et al. (2005) [18] avec le même niveau de sécurité ; | | - Vulnérable à plusieurs types d'attaques ;<br>- le concept de session alourdit la charge de stockage | |
| **Chuang et al. (2010) [102]** | - résistant aux attaques sybil, et aux attaques via les clés de chemin ;<br>- combine les mérites de la crypto sym. et asym. | | - requiert que les horloges du système soient synchronisées (usage d'horodatage contre les attaques par rejeu) ;<br>- ne précise pas comment le message de révocation qui termine le processus sera authentifié. | |

| | | | | |
|---|---|---|---|---|
| **Moore et al. (2007) [27]** | - résistant aux attaques sybil, aux attaques via les clés de chemin et aux attaques par fausse accusation (basé sur le vote positif) | | - Vulnérable aux attaques DoS (certains nœuds peuvent refuser de fournir les services de mise à jour de jeton) ;<br>- moins performant que tous les autres protocoles distribués. | nœuds du réseau ;<br>- Construit sur des hypothèses simplificatrices (connaissance préalable de la position post-déploiement, les nœuds sont immobiles, présence de fonctionnalités pour la réplication des nœuds et Sybil |

**Tableau 5. 19 : Avantages et Inconvénients des Protocoles Hybrides**

| **Protocoles Hybrides** | **AVANTAGES** | | **INCONVENIENTS** | |
|---|---|---|---|---|
| **Sanchez et Baldus (2005) [103]** | - seul protocole hybride qui prend en compte la mobilité | - adressent tous certaines failles de sécurité rencontrées avec les protocoles distribués notamment la vulnérabilité aux attaques sybil, attaques par réplication, attaques via les clés de chemin) ;<br>- ont pratiquement les mêmes performances que les protocoles distribués, mais sont plus résistants que ces derniers aux potentielles attaques. | - les auteurs ne donnent pas assez de détails sur le fonctionnement du protocole. | - tous sont basés sur une hypothèse simplificatrice : l'autorité centralisée est hors de portée des adversaires ;<br>- existence d'un unique point de défaillance ;<br>- Absence de mécanisme de gestion de la confiance pour contrer les abus (la majorité ne parvient pas à contrer les attaques par fausse accusation et les attaques par coalition de nœuds) ; |
| **Chattopadhyay et Turuk (2012) [106]** | - répond à plusieurs insuffisances de Chan et al. (2005) [18] ;<br>- élimine l'hypothèse de la connaissance préalable de la position post-déploiement ;<br>- consomme moins de ressources de stockage et de calcul. | | - se base sur une hypothèse simplificatrice pour résister aux attaques par coalition de nœuds (moins de $t_r$ nœuds compromis dans le voisinage d'un nœud) ; | |
| **Jiang et al. (2013) [107]** | - bien que basé sur le vote simple avec seuil, l'implication du nœud CH permet une résistance face à plusieurs types attaques **(si et seulement si le nœud CH est non compromis)** | | | |

| | | | | |
|---|---|---|---|---|
| **Ge et Choo (2014) [109]** | - élimine l'hypothèse de la connaissance préalable de la position post-déploiement ; | | - vulnérable aux attaques sybil (suppose juste que les mécanismes pour contrer ce type d'attaques sont en place;<br>- moins performant que les autres protocoles (consomme plus de ressources de stockage et de calcul) ;<br>- propose la complète notification, mais ne traite pas les problèmes associés (problème de l'authentifi-cation par exemple) | |

## 5.4 Discussions et Conclusion

Dans ce chapitre, nous avons réalisé une étude évaluative et comparative des différentes catégories de protocoles proposés dans le cadre de la révocation de clés dans les WSNs en nous basant sur des critères relatifs à leur sécurité, leur efficacité et leur applicabilité. Cette étude nous permet de constater que :

- En ce qui concerne l'aspect sécurité, les solutions centralisées présentent une plus grande robustesse face aux potentielles attaques (voir tableau5.1), mais il est à noter que la sécurité de ce type de protocoles dépend entièrement de celle de l'autorité de confiance impliquée. Pour cette catégorie de protocoles, les auteurs écartent toute possibilité de compromission de celle-ci en supposant juste que cette dernière est hors de portée des adversaires. Cependant, la vérification de cette hypothèse ne peut pas être garantie dans toutes les applications des WSNs, d'où l'existence pour ces protocoles d'un unique point de défaillance ;
- En ce qui concerne l'aspect efficacité ou performance, le tableau 5.4 et le tableau 5.6 montrent que les protocoles centralisés présentent des atouts en termes de consommation des ressources de stockage et de calcul des nœuds simples car la complexité associée à ces paramètres vaut $O(1)$ pour presque tous les protocoles. Ceci est dû à l'implication d'une autorité de confiance dont le rôle est d'exécuter presque toutes les tâches requises par le processus de révocation. Cependant pour ce type de protocole ; lorsque nous nous intéressons à la charge de communication, à la charge globale de calcul, et au délai ou latence de la révocation, nous constatons que les protocoles centralisés sont les moins performants car les complexités liées à ces paramètres sont dépendantes de la taille du réseau $n$, voir le tableau 5.5, le tableau 5.6 et le tableau 5.7 ;

- Tous les protocoles centralisés, excepté [92], ne posent pas de contraintes sur la mobilité des nœuds participants, mais ne supportent pas le passage à l'échelle. Ce qui les rend moins flexibles et restreint leur utilisation aux applications ne nécessitant pas le déploiement d'un grand nombre de nœuds ;
- En proposant des solutions qui évitent l'implication d'une quelconque autorité de confiance centralisée, les protocoles distribués éliminent l'existence d'un unique point de défaillance, et tous, excepté [27], adressent les limites des solutions centralisées liées à leur performance. Ces solutions distribuées présentent des charges de communication, des charges globales de calcul et un délai de révocation de complexité moins élevées, non dépendante de la taille du réseau, mais plutôt du nombre de nœuds participants $m$ et/ou du seuil de révocation $t_r$ et/ou du nombre total de sauts $k$ à parcourir dans le réseau. Lorsque ces grandeurs sont constantes nous obtenons pour chacun des paramètres de performance précités une complexité constante égale à $O(1)$. Il est à noter cependant que la non implication d'une autorité centralisée ne permet pas pour autant à cette deuxième catégorie de solution distribuée de répondre à tous les problèmes de sécurité et de performance. En effet, l'entière distribution des tâches du processus de révocation aux différents nœuds simples du réseau et l'absence de mécanisme pour contrer les éventuels abus rend les solutions distribuées plus complexes et vulnérables à plusieurs types d'attaques de sécurité (voir tableau 5.2). L'approche distribuée crée également au niveau des nœuds participants une plus grande consommation de ressources de stockage et de calcul (voir Tableau 5.8 et tableau 5.10). En plus de ces limites, s'ajoutent celles qui sont liées à leur applicabilité (voir tableau 5.16) ;
- L'évaluation des protocoles hybrides montre qu'en combinant l'approche centralisée et celle distribuée ceux-ci bénéficient des avantages de chacune d'elles, et apportent des solutions à certaines des failles de sécurité rencontrées avec les protocoles distribués (voir tableau 5.3) tout en gardant pratiquement les mêmes valeurs de performance. Cependant les résultats mentionnés dans les tableaux 5.2 et tableau 5.3 montrent que cette nouvelle approche hérite également des faiblesses, ou problèmes majeurs, rencontrées avec les approches antérieures, notamment la vulnérabilité aux fausses accusations, aux attaques par coalition de nœuds (approche basée simplement sur le vote), et pour couronner le tout l'existence d'un unique point de défaillance.

# Chapitre 6 PROPOSITION DE SOLUTION POUR LA REVOCATION DE CLÉS DANS LES MANETs

Dans ce chapitre nous présentons la solution que nous avons proposée dans le but d'adresser les insuffisances et limites soulignées dans le chapitre 4 concernant la révocation de clés dans les MANETs.

## 6.1 Contexte et Motivation

L'analyse des résultats du chapitre 4 montre que chaque catégorie de protocoles de révocation de clés proposée pour les MANETs présente des forces et des faiblesses liées à sa sécurité à son efficacité et/ou à son applicabilité. En effet :

- En ce qui concerne l'aspect sécurité, les protocoles décentralisés ainsi que les premières propositions de solutions distribuées uniquement basées sur le vote avec seuil, notamment [6] et [8] ne sont pas assez robustes pour offrir un niveau convenable de sécurité en raison de leur vulnérabilité aux attaques sybil, aux attaques par réplication de nœuds et surtout aux attaques par révocation de nœud. Ce qui constitue une faille de sécurité majeure dans le domaine de la révocation de clés. Les solutions de la catégorie distribuée proposées par la suite, ainsi que celles de la catégorie hybride, adressent ce problème en renforçant leur protocole par un mécanisme de gestion de la confiance, ce qui a permis un rehaussement de leur niveau sécurité ;
- Pour ce qui est de l'aspect efficacité, les protocoles décentralisés, ainsi que certains protocoles distribués, présentent des limites en termes de performance. Cela est dû à l'implication de l'ensemble des nœuds du réseau dans le processus de révocation d'un seul nœud malicieux, ou au stockage au niveau de chaque nœud d'une table avec $n$ entrées, renseignant sur le statut de tous les autres nœuds dans le réseau. Tout comme les protocoles décentralisés, les protocoles hybrides présentent également des limites, puisque leur concepteurs ne fournissent pas les détails requis pour évaluer leur efficacité et pouvoir les comparer aux autres types de protocoles ;
- Lorsque nous nous intéressons à leur applicabilité, nous constatons que les protocoles décentralisés et quelques-uns parmi les protocoles distribués peinent à trouver leur applicabilité dans les réseaux MANETs de grande taille puisqu'ils présentent des charges de stockages et des charges globales de calcul, dont la complexité dépend de la taille $n$ du réseau. Avec les protocoles hybrides, l'implication d'autorités centralisées dans le mécanisme associé de gestion de la confiance réduit leur champ d'application uniquement aux réseaux MANETs planifiés.

Il apparait alors que les protocoles de la catégorie distribuée présentent le plus d'atouts lorsqu'ils sont comparés aux autres types de protocoles car :

- certains parmi eux fournissent des solutions qui adressent les vulnérabilités des autres protocoles et parviennent à atteindre un niveau supérieur de sécurité ;
- certains parviennent à réduire les charges de stockage, de calcul, et les charges de communication, puisque les complexités associées ne dépendant plus de la taille $n$ du réseau.
- leur caractère distribué fait d'eux des candidats qui répondent le mieux au caractère auto-organisé des MANETs. De plus, en étant plus performants, certains parmi eux éliminent les restrictions qui se posaient aux autres protocoles quant à leur applicabilité aux réseaux MANETs de grande taille.

Malgré l'existence d'atouts en termes de sécurité, d'efficacité et d'applicabilité, les solutions distribuées ne résolvent pas pour autant tous les problèmes soulignés lors de notre analyse dans le chapitre 4. Pour cette raison, dans ce chapitre, nous nous concentrons sur la catégorie de protocoles distribués en nous focalisant sur certains des problèmes non encore résolus et en tentant d'y apporter des solutions.

En analysant de plus près les résultats donnés, dans le chapitre 4, pour les protocoles de la catégorie distribuée, nous constatons que les problèmes liés à la sécurité sont principalement dus à leurs vulnérabilités aux attaques par exhaustion de batterie, aux attaques par fausse accusation, aux attaques sybil, aux attaques par coalition de nœuds, et quelquefois aux attaques par itinérance. Nous avons également pu constater que ces insuffisances se font beaucoup plus sentir avec les protocoles basés sur PKI, puisque les premières propositions de cette catégorie sont sujettes à plusieurs des attaques précitées (attaques par fausse accusation, attaques sybil, attaques par coalition de nœuds). Malgré l'existence de solutions plus récentes pour cette catégorie basée sur PKI [82], [83], nous notons qu'aucune ne parvient à dresser une ligne de défense qui lui permet de résister à toutes les attaques potentielles considérées dans notre étude.

Pour ce qui est des problèmes liés à leur efficacité nous notons qu'ils sont généralement dus à un manque de mécanisme efficace pour assurer l'authentification du transfert de messages par broadcast et/ou à un manque de mécanisme efficace pour la gestion de la confiance. Pour ce problème également, le constat est que la catégorie de protocoles basés sur PKI est la plus touchée, puisqu'aucun de ces protocoles n'y répond efficacement.

**Tableau 6. : Sécurité et Efficacité des Protocoles Distribués – Synthèse de l'analyse**

| Protocoles Distribués (MANETs) | | SÉCURITÉ | EFFICACITÉ | |
|---|---|---|---|---|
| | | Résistance à tous les potentiels abus considérés | Efficacité de l'authentification du Broadcast | Efficacité du mécanisme de gestion de la Confiance |
| PKI | Luo et al. (2004) [6] | Non | Non | Non |
| | Arboit et al. (2008) [77] | Non | Non | Non |
| | Chaib et al. (2012) [82] | Non | Non | Non |
| | Guo et al. (2014) [83] | Non | Non | Non |
| IDENTITÉ | Saxena et al. (2004) [8] | Non | Non | Non |
| | Zhang et al. (2006) [10] | Non | Non | Non |
| | Fan et al. (2008)[79] | Non | Non | Non |
| | Hoeper et al. (2009) [28] | Oui | Non | Oui |

Le tableau 6.1 synthétise les résultats donnés dans le chapitre 4 et mentionnés ci-dessus. D'après celui-ci, seul le protocole décrit dans [28] (que nous appelons par Hoeper et Gong (2009) dans la suite) de la catégorie des propositions basées sur l'identité propose une solution qui répond à certains des problèmes mentionnés ci-dessus. Pour cette raison, nous proposons dans [118], le protocole de révocation de certificat nommé SECRET (*SECRET : A Secure and Efficient Certificate REvocation scheme for Mobile Ad hoc NeTworks*), qui adresse les insuffisances relatives aux protocoles basés sur PKI par la fourniture d'un mécanisme qui se base sur le protocole Hoeper et Gong (2009), et qui utilise le protocole d'authentification du broadcast fourni par [119].

## 6.2 Fondements de SECRET

### 6.2.1 Choix des Protocoles de Référence

Parmi les solutions de révocation distribuées proposées pour les MANETs, celle décrite dans [28] semble être la plus efficace, prenant en compte les vulnérabilités et manquements des autres solutions. Ceci parce qu'elle :

- **Combat les abus en :**
  - Sécurisant les messages d'accusation ;

- Accompagnant le protocole par un mécanisme de gestion de la confiance assez robuste, qui prend aussi en compte les inexactitudes relatives au système de surveillance implémenté au niveau des nœuds ;

- **Assure un niveau acceptable de performance en :**
  - Sécurisant les messages d'accusation, non pas par le biais de signature, mais avec les MACs (*Message Autentication Code*) calculés via les paires de clés générées de façon non interactive ;
  - Proposant un mécanisme de gestion de la confiance efficace, qui prend en charge efficacement l'arrivée de nouveaux nœuds dans le réseau, et ne produit pas une lourde charge de stockage (limitation du nombre de participants, utilisation de listes de révocation de clés sous format matriciel, stockant l'historique des accusations de chaque nœud participant).

Toutefois, il est à noter que malgré tous ses atouts sur les autres protocoles distribués, Hoeper et Gong (2009) peine à atteindre un bon niveau d'efficacité en raison du transfert par unicast pour la propagation de messages d'accusation. En fait, étant basé sur l'identité, Hoeper et Gong (2009) dispose uniquement de paires de clés pour assurer la protection des échanges, et pas d'une clé unique pouvant assurer l'authentification du broadcast, d'où l'accroissement de la charge de communication et du délai de révocation en cas d'accroissement du nombre de nœuds participants au processus de révocation de clés d'un nœuds compromis.

Afin de répondre aux insuffisances des protocoles basés sur PKI en leur faisant bénéficier des atouts du protocole Hoeper et Gong (2009), nous devons au préalable trouver un moyen de réduire autant que possible la charge de communication et le délai de révocation associés. Pour ces raisons, nous proposons d'utiliser le protocole d'authentification du broadcast *HEAP (Hop-by-hop Efficient Authentication Protocol)* [119], [120] afin d'adapter Hoeper et Gong (2009) aux systèmes basés sur PKI.

Nous optons pour le protocole HEAP comme protocole d'authentification, car dans [120] celui-ci à été comparé aux autres protocoles d'authentification du broadcast tels que Tesla [121-123], et a été prouvé comme étant plus sécurisé et plus performant. De plus ce protocole répond bien à nos besoins, puisqu'il permet une authentification saut par saut lors de la propagation des messages d'accusation.

### 6.2.2 Description du protocole Hoeper et Gong (2009)

Dans [28], Hoeper et Gong présentent un protocole de révocation de clés auto-organisé. Les auteurs introduisent un nouveau type de schéma à clé publique, basé sur l'identité d'un nœud, la date d'expiration de la clé et sa version. Dans leur proposition, chaque nœud utilise un mécanisme nommé *Neighborhood watch* pour surveiller les nœuds de son voisinage à un saut. En cas de détection d'un comportement malveillant, ses observations sont propagées vers le voisinage à $m$ sauts de façon sécurisée via les MACs calculés à l'aide de paires de clés partagées entre les nœuds communiquants, la protection des messages d'accusation étant assurée par les clés de paire obtenues de façon non interactive via un protocole d'accord de clés basé sur l'identité. De plus quand le nœud réalise que sa clé privée a été compromise, il génère un message *Harakiri* et le propage à travers tout le réseau. Un nœud $A$ considère la clé publique d'un nœud $B$ comme révoquée si au moins une des trois conditions suivantes est satisfaite :

- $A$ observe le comportement malveillant de $B$ à travers le *Neighborhood watch* ;
- $A$ reçoit de $B$ un message *Harakiri* déclarant que sa clé privée est compromise ;
- $A$ reçoit au moins $\delta$ accusations contre le nœud $B$ en provenance de nœuds fiables de son voisinage à $m$ sauts, $\delta$ étant le seuil de révocation prédéterminé.

Pour mitiger l'influence des attaques par fausse accusation des nœuds coalisants à $l$ sauts, $(2 \leq l \leq m)$, le vote majoritaire avec le paramètre $\varepsilon$ est en plus appliqué. Les nouveaux nœuds peuvent simplement rejoindre le réseau et participer aussitôt à un processus de révocation de clé sans avoir à vérifier au préalable un grand nombre d'accusations antérieures comme le font les protocoles [77] et [83]. Quatre algorithmes composent la solution basique proposée par Hoeper et Gong :

1. Algorithme 1 : *Neighborood Watch ;*
2. Algorithme 2 : *Harakiri ;*
3. Algorithme 3 : *Propagate* ;
4. Algorithme 4 : *Update KRL (Key Revocation List).*

La solution suggérée par Hoeper et Gong présente plusieurs atouts, mais n'est pas directement applicable aux systèmes utilisant PKI. Ceci est principalement dû à deux facteurs :

- l'usage de clés publiques auto authentifiées ;
- l'usage de clés de paire pré-partagées pour sécuriser les messages d'accusation.

En effet, avec les protocoles de révocation de clés basés sur l'identité proposés pour les MANETs, les clés publiques des utilisateurs et leur identité ne dépendent pas des certificats pour prouver leur authenticité, ce qui n'est pas le cas avec les autres types de protocole où cette authenticité requiert

d'être établie via le certificat associé. Une fois que ce certificat est délivré il devient indispensable de révoquer la clé publique d'un nœud en passant par la révocation de son certificat. De plus, la solution présentée par Hoeper et Gong utilise des clés de paire $k_{i,j}$ pour sécuriser les messages d'accusation et ces dernières sont établies par le biais d'une opération de couplage (*pairing*) de façon non interactive selon la formule (6.1).

$$k_{i,j} = \hat{e}(d_i, Q_j) = \hat{e}(Q_i, d_j) \qquad (6.1)$$

où $_i$, $d_j$ sont les clés privées, $Q_i$, $Q_j$ sont les clés publiques des nœuds respectifs $i$ et $j$ et ê est la fonction de correspondance bilinéaire (*bilinear map).*

### 6.2.3 Description du protocole d'authentification HEAP

Dans la phase de démarrage de HEAP [119], un nouveau nœud $i$ génère en premier lieu une clé unique nommée $ikey$ et une clé de paire pour chaque voisin d'un saut $j$ nommée $okey$. Les clés sont des chaînes de bits aléatoires de taille fixe faciles à générer. Pour ce faire, le nœud $i$ utilise un mécanisme d'échange de clés standard, par exemple le mécanisme d'échange de clés *Diffie-Hellman*, utilisant la cryptographie à courbes elliptiques pour pouvoir partager en toute sécurité la clé de groupe $ikey$ avec les voisins correspondants. Chaque fois que le voisinage d'un nœud change en raison de la mobilité, $i$ partage une nouvelle clé de groupe et une nouvelle clé de paire avec chaque nouveau voisin. Après la phase de génération et de distribution des clés, l'authentification des paquets peut avoir lieu.

La phase d'authentification peut être décrite comme suit : quand un nœud $i$ veut diffuser un message à tous ses voisins d'un saut, en permettant à chacun d'eux d'authentifier l'origine du paquet, il peut procéder en générant un nouveau HMAC [66] pour chaque voisin $j$ en utilisant la clé de paire $okey_j$. Cependant, ce calcul serait trop coûteux (voir la figure 6.1 suivante).

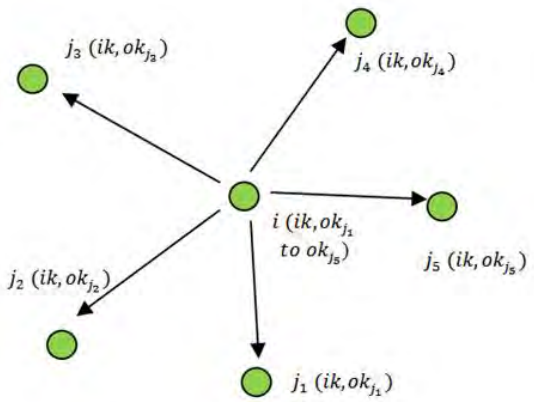



**Figure 6. : Le nœud $i$ doit transmettre un paquet à ses voisins $j_1$ à $j_5$.**

Pour réduire le coût de calcul, les auteurs ont proposé un algorithme plus efficace utilisant un HMAC légèrement modifié. Le HMAC original est calculé par :

$$HMAC(M,k) = H(k\ XOR\ opad \mid H(k\ XOR\ ipad \mid M) \qquad (6.2)$$

$H(x)$ étant une fonction cryptographique de hachage, $M$ le message à transmettre, $'opad'$ est le nombre hexadécimal 5C utilisé pour remplir chaque octet de $k$ jusqu'à une taille de bloc. $'ipad'$ est le nombre hexadécimal 36 utilisé pour remplir chaque octet de $k$ jusqu'à une taille de bloc (c'est-à-dire 512 bits). Le symbole '|' représente la concaténation.

Le calcul du HMAC selon la formule (6.2) peut se faire comme suit :

Étape1 : $H(k\ XOR\ ipad \mid M)$
Étape 2 : $H(k\ XOR\ opad \mid hachage\ de\ l'etape\ 1)$

Pour les messages longs, l'étape 1 entraînera la plus grande partie de la surcharge de calcul. Cette idée a conduit les concepteurs à modifier l'algorithme pour les applications de diffusion par broadcast en utilisant deux clés $ikey$ et $okey$. La première clé $iKey$ est rembourrée (complétée) avec des 0 pour en faire une taille de bloc avant d'être utilisée pour générer le hachage dans la nouvelle étape 1 suivante :

Étape1 : $H(ikey \mid M)$

Notons que l'expéditeur n'a besoin de calculer cette étape coûteuse qu'une seule fois, car tous les voisins à un saut partagent la même clé $ikey$. Pour éviter l'usurpation d'identité du nœud $i$ par l'un de ses voisins à un saut, la clé de paire $okey$ est utilisée pour générer le hachage à la nouvelle étape 2 suivante. $okey_j$ est complétée avec des 0 pour en faire un bloc avant d'être utilisée.
Étape 2 : $H\big(okey_j \big| H(ikey \mid M)) = MAC_j$

Cette étape doit être calculée une fois pour chaque voisin à un saut $j$. Pour se protéger contre les attaques par rejeu, un numéro d'index est inclus dans le paquet avant la transmission. Le numéro d'index est concaténé avec les données utiles pour donner le message $M$. L'index est incrémenté de un pour chaque message suivant. Le format de chaque paquet envoyé par un nœud $i$ est alors :

$$i \rightarrow * : M, index, MAC_1(M \mid index), \ldots, MAC_n(M \mid index)$$

Après réception de ce message, un nœud voisin à un saut $j$ calcule le MAC pour voir s'il correspond à l'une quelconque des étiquettes MAC dans le message et si le numéro d'index est valide. Si c'est le cas, le message est accepté comme authentique ; sinon il sera ignoré.

## 6.3 Adaptation de la solution de Hoeper aux systèmes basés sur PKI

Nous considérons que le système pour lequel nous proposons une solution de révocation est basé sur PKI avec une autorité de certification (CA) *off-line* chargée d'initialiser les nœuds du réseau avant leur déploiement. Chaque nœud pourra alors communiquer avec cette autorité de certification approuvée avant de rejoindre le réseau afin obtenir un certificat de clé publique unique signé par l'autorité de certification, ainsi que la clé publique authentique de l'autorité de certification.

Nous considérons également que tous les liens de communication directs entre les nœuds sont bidirectionnels, et que chaque nœud a un système de surveillance mis en œuvre. Ceci est nécessaire pour permettre aux nœuds de surveiller leurs voisins d'un saut. Nous supposons enfin que chaque nœud connaît ses voisins à un saut. Cela est nécessaire pour assurer une distribution complète des clés partagées $ikey$ et $okey_j$ mentionnées dans la section précédente.

### 6.3.1 Description de SECRET

Les symboles que nous utilisons dans la suite sont listés dans le tableau 6.2

**Tableau 6. : Symboles et Notations utilisés**

| | |
|---|---|
| $n$ | nombre total de nœuds dans le réseau |
| $N_{1,i}, \sigma_i$ | ensemble et nombre de nœuds dans le voisinage d'un saut du nœud $i$ |
| $N_i, \Omega_i$ | la perception du nœud $i$ de l'ensemble des nœuds de son voisinage à $m$ sauts et du nombre de nœuds dans celui-ci. |
| $\delta$ | seuil de révocation |
| $\varepsilon$ | seuil pour le vote majoritaire (pour l'acceptation des accusations rapportées) |
| $\alpha, \beta$ | proportions respectives de faux positifs et de faux négatifs du mécanisme de surveillance |
| $m$ | portée de propagation des accusations |

Nous présentons ici SECRET, qui est une version modifiée du mécanisme de révocation proposé par Hoeper et Gong dans [28]. Il comporte trois algorithmes qui fonctionnent de manière similaire à ceux décrits dans [28].

Afin d'adapter la proposition de référence aux systèmes basés sur PKI, nous utilisons :

*i)* les listes de révocation de certificat à la place de listes de révocation de clés ;

*ii)* HEAP comme protocole d'authentification. Les clés $ikey$ et $okey$ sont alors utilisées pour produire les MACs et assurer l'authentification des messages d'accusation, tel que décrit dans la section II.3.

*iii)* le broadcast à la place de l'unicast pour la propagation des messages d'accusations ;

#### 6.3.1.1 Création de listes de révocation de certificat

Par analogie aux tables de profils et/ou tables de statuts décrites par les protocoles distribués basés sur PKI tels [77] et [83], dans [28], Hoeper et Gong proposent d'établir pour chaque nœud $i$ du réseau une liste de révocation de clés (*Key Révocation List (*$KRL^i$)). Pour limiter l'espace requis pour le stockage de celle-ci tout en incluant dans cette dernière les renseignements permettant vérifier le statut et la fiabilité d'un nœud voisin, les auteurs adoptent le format matriciel et limitent le nombre d'entrées (nombre de lignes de la matrice) au nombre de nœuds connus dans le voisinage à $m$ sauts. En nous inspirant de ces listes de révocation de clés, nous créons pour chaque nœud $i$ du réseau une liste de révocation de certificat $CRL^i$ (*Certificate Révocation List* ($CRL^i$)) pour le stockage des informations relatives à la fiabilité et au statut des nœuds appartenant au voisinage à $m$ sauts. Comme $KRL^i$, la liste $CRL^i$se présente sous forme matricielle avec une dimension $(\Omega_i \times \Omega_i + 3)$ de la forme :

$$CRL^i = \begin{pmatrix} a^i_{1,1} & \dots & a^i_{1,j} & \dots & a^i_{1,\Omega_i} & cert_1 & v^i_1 & X^i_1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a^i_{j,1} & \dots & a^i_{j,j} & \dots & a^i_{j,\Omega_i} & cert_j & v^i_j & X^i_j \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a^i_{\Omega_i,1} & \dots & a^i_{\Omega_i,j} & \dots & a^i_{\Omega_i,\Omega_i} & cert_{\Omega_i} & v^i_{\Omega_i} & X^i_{\Omega_i} \end{pmatrix} \qquad (6.3)$$

Dans cette liste, les valeurs d'accusation sont représentées par $a^i_{k,j} \in \{0,1\}$ avec $\{k,j\} \in \{1,\dots,\Omega_i\}$. Notons que $CRL^i$ doit être ajustée chaque fois que $\Omega_i$ change, c'est-à-dire une nouvelle colonne doit y être insérée pour chaque nouveau nœud dans $N_i$. La valeur $a^i_{k,j}$ indique que le nœud $i$ a "entendu " que le nœud $j$ accuse le nœud $k$ de comportement malveillant ($a^i_{k,j} = 1$) ou croit que $k$ est digne de confiance ($a^i_{k,j} = 0$). L'indice supérieur $i$ dénote que les valeurs sont des valeurs actuelles dans $CRL^i$ de $i$.

Chaque $j$-ième vecteur colonne $\underline{c}^i_j$ dans $CRL^i$, $1 \le j \le \Omega_i$ contient toutes les accusations $a^i_{k,j}$ portées par le nœud $j$ contre les nœuds $k \in N_i$. Chaque $j$-ième vecteur ligne $\underline{r}^i_j$ dans $CRL^i$, $1 \le j \le \Omega_i$ correspond à un nœud $j \in N_i$ et contient, entre autres informations, les valeurs d'accusation $a^i_{j,k}$ de tous les nœuds $k \in N_i$ évaluant le nœud $j$. En particulier, les éléments de 1 à $\Omega_i$ de la ligne $\underline{r}^i_j$ contiennent les valeurs d'accusation de $a^i_{j,1}$ à $a^i_{j,\Omega_i}$. L'élément $(\Omega_i + 1)$ contient le numéro de série $Cert_j$ du certificat du nœud $j$, l'élément $(\Omega_i + 2)$ représente la période de validité $v^i_j$ du certificat du nœud $j$, et L'élément $(\Omega_i + 3)$ contient un indicateur $X^i_j$ qui, lorsque fixé à 1, indique que le nœud $i$ considère le certificat du nœud $j$ comme révoqué.

Le nœud $i$ fixe

$$X_j^i = \begin{cases} 1 & si \quad a_{j,i}^i = 1 \quad (condition1) \\ & ou\ si \quad \sum_k a_{j,k}^i \geq \delta\ \forall k \in \Omega_i\ avec\ X_k^i = 0 \quad (condition2) \\ 0 & sinon \end{cases} \qquad (6.4)$$

Fondamentalement, le nœud $i$ considère le certificat du nœud $j$ comme étant révoqué si au moins l'une des conditions 1 ou 2 est satisfaite. La condition 1 décrit le cas où le nœud $i$ observe le comportement malveillant du nœud $j$ lors de son *Neighborhood Watch.* La condition 2 concerne le cas pour lequel le nœud $i$ reçoit au moins $\delta$ accusations contre le nœud $j$ en provenance de nœuds $k \in \Omega_i$ avec $X_k^i = 0$. Si aucune de ces deux conditions n'est satisfaite, le nœud $i$ considère le nœud $j$ et son certificat actuel comme fiables, c'est-à-dire $X_j^i = 0$.

#### 6.3.1.2 Détails du Protocole

Nous présentons ici une solution permettant de révoquer uniquement les clés compromises des autres nœuds voisins. Ainsi, à la différence du protocole [28] qui se compose de quatre algorithmes, notre mécanisme se composera de trois :

1. Algorithme 1 : *Neighborood Watch* ;
2. Algorithme 2 : *Propagate ;*
3. Algorithme 3 : *Update CRL (Certificate Révocation List)*.

Leur exécution s'enchaine selon la figure 6.2 donnée ci-après.

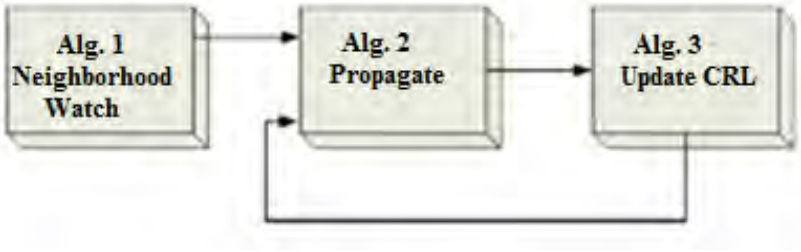



**Figure 6. : Description du fonctionnement de notre solution**

**Algorithme 1 : *Neighborhood Watch***

Dans cet algorithme, chaque nœud $i$ surveille ses voisins à un saut. Chaque fois qu'un nœud $i$ observe un voisin suspect $j \in N_{1,i}$, il fixe $a_{j,i}^i = 1$ et crée un *neighborhooh watch* message $nw_i$ sous ce format :

$$nw_i \rightarrow * : M, MAC_1, MAC_2, \ldots, MAC_{\sigma_i}$$

Avec :

- $*$ pour signifier tout destinataire parmi les voisins directs du nœud $i$ ;
- $M$ pour signifier le message d'accusation comprenant le numéro de série du certificat $Cert_i$ du nœud accusateur $i$, la liste de révocation de certificat $CRL^i$ du nœud $i$ et le $hopcount$, qui est une valeur indiquant le nombre de sauts que le message doit encore parcourir pour atteindre tout le voisinage à $m$ sauts. Nous avons inclus dans le message la liste complète de révocation de certificat. Cependant comme l'a suggéré Hoeper et Gong, nous pourrons optimiser l'usage de la bande passante en n'incluant dans celui-ci que les vecteurs de cette liste de révocation mis à jour ;
- $MAC_1, MAC_2, \ldots, MAC_{\sigma_i}$ pour signifier les différents $MACs$ calculés pour chaque voisin d'un saut $j \in N_{1,i}$ selon l'étape 2 décrite à la section II.3.

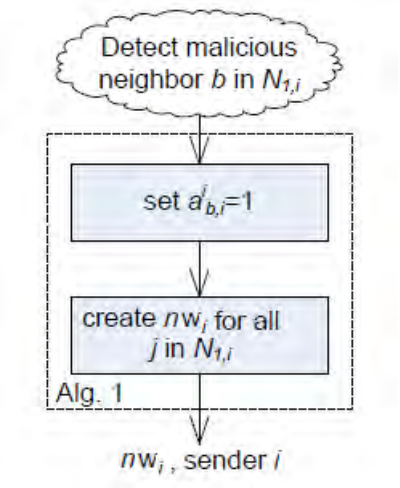



**Figure 6. : Organigramme de l'algorithme N*eighborhood Watch***

Après génération d'un *Neighhborhood Watch* message, le nœud $i$ utilise l'Algorithme 2 pour le propager.

**Algorithme 2 : *Propagate***

Cet algorithme est déclenché soit par l'algorithme 1 soit par l'algorithme 3. Après création d'un message d'accusation $am_i$ qui peut être soit un message *Neighhborhood Watch* $nw_i$ soit un message de mise à jour de liste de révocation $um_i$ (*update message)* les nœuds propagent de façon sécurisée ces accusations vers leur voisinage à un saut.

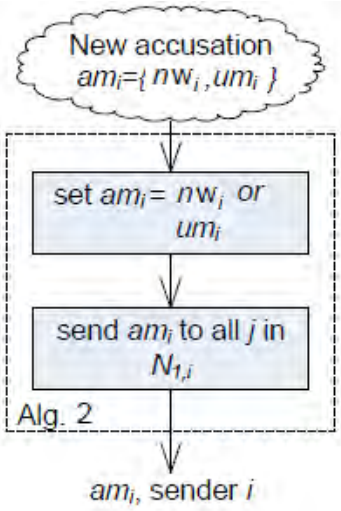



**Figure 6. : Organigramme de l'algorithme *Propagate***

**Algorithme 3 : *Update CRL***

Cet algorithme décrit comment le nœud $i$ met à jour sa propre liste de révocation de certificat $CRL^i$ en concordance avec le type de message d'accusation $am_j$ reçu.

*Message reçu* $am_j = nw_j$

Lorsque le nœud $i$ reçoit un message de surveillance de voisinage $nw_j$, il doit exécuter plusieurs étapes de vérification et de mise à jour. Si une étape réussit, $i$ continue avec l'étape suivante, sinon $i$ laisse tomber le paquet et abandonne l'algorithme. A la réception de $am_j = nw_j$, le nœud $i$ effectue les étapes suivantes :

1. *vérification de voisin :* $i$ vérifie si l'expéditeur $j$ est un voisin direct, c'est-à-dire si $j \in N_{1,i}$ ;

2. *vérification de la fiabilité :* $i$ vérifie s'il fait confiance à $j$, c'est-à-dire $X_j^i = 0$ ;

3. *vérification de l'authenticité du message :* $i$ vérifie la correspondance de MAC du message reçu $nw_j$ en utilisant les clés $ikey$ et $okey$, tel que décrit par le protocole d'authentification HEAP ;

4. e*xtension de la liste de révocation de certificat* : le nœud $i$ scrute $CRL^j$ pour les lignes associées aux nœuds $k$ tel que $k \notin N_i$ et pour tout tel $k$ crée une nouvelle ligne $\underline{r}_k^i$ dans $CRL^i$, définit $N_i := N_i + ID_k$, $ID_k$ étant l'identité du nœud $k$ et $\Omega_i = \Omega_i + 1$ ;

5. *copie des accusations directes* : $i$ extrait le vecteur de colonne $\underline{c}_j^j$ de $nw_j$ pour mettre à jour son propre vecteur de colonne $\underline{c}_j^i$ dans $CRL^i$. Pour la mise à jour, le nœud $i$ copie uniquement les valeurs d'accusation $a_{k,j}^j = 1$ pour n'autoriser que les transitions de 0 à 1. Après avoir terminé, le nœud $i$

définit la mise à jour, c'est-à-dire la variable *update* signifiant que la mise à jour est fixée à vraie (*update = true*) ;

6. *stockage des accusations rapportées* : $i$ scrute toutes les colonnes $\underline{c}_k^j$ avec $k \in \{1, \dots, \Omega_i\}$ dans $CRL^j$ et stocke toutes les colonnes $\underline{c}_k^j$ pour lesquelles toutes les conditions suivantes sont satisfaites :

(a) $k \neq i$

(b) $k \neq j$

(c) $X_k^i = 0$

(d) $k \notin N_{1,i}$.

Le nœud $i$ vérifie les conditions (a) - (d) pour tout $k \in \{1, \dots, \Omega_i\}$. Si toutes les conditions sont remplies pour $k$, $i$ stocke $\underline{c}_k^j$ et incrémente le compteur $r_k$ pour les $k$-vecteurs reçus. Toutes les autres colonnes sont ignorées. Nous avons besoin de recevoir au minimum $\varepsilon$ $k$-vecteurs de différents voisins à un saut pour établir la confiance dans les accusations signalées du nœud $k$. Si au moins $\varepsilon$ $k$-vecteurs sont collectés, $i$ met à jour son $CRL^i$ comme décrit dans l'étape suivante ;

7. *Utilisation des k-vecteurs accumulés pour la mise à jour* : pour tout nœud $k$, $k \in \{1, \dots, \Omega_i\}$, soit $r_k$ le nombre de vecteurs colonnes $\underline{c}_k^j$ reçus et stockés à l'étape 6 par le nœud $i$ à partir de $r_k$ voisins à un saut $j$, avec$j \in \{1, \dots, r_k\}$ pour une représentation plus facile et sans perte de généralité. Pour tout nœud $k$, $k \in \{1, \dots, \Omega_i\}$, le nœud $i$ vérifie si $r_k \geq \varepsilon$. Si c'est le cas, le nœud $i$ met à jour le $k$-vecteur dans $CRL^i$. Chaque valeur d'accusation $a_{l,k}^i$ avec $l \in \{1, \dots, \Omega_i\}$ dans $\underline{c}_k^i$ est calculée à partir du vote majoritaire sur tous les $a_{l,k}^j$ recueillies, avec

$$a_{l,k}^i = \begin{cases} 1 & si \ \sum_{j=1}^{r_k} a_{l,k}^j \geq \frac{r_k}{2} \\ a_{l,k}^i & sinon \end{cases} \qquad (6.5)$$

où $j \in \{1, \dots, r_k\}$. Notons que tous les $k$-vecteurs qui ont été utilisés pour la mise à jour de *CRL* sont effacés, tandis que les $k$-vecteurs non utilisés, c'est à dire ceux pour lesquels $r_k < \varepsilon$, restent dans le stockage de $i$. Si $i$ met à jour au moins un $k$-vecteur dans $CRL^i$, le nœud $i$ fixe l'indicateur de mise à jour à vraie (*update = true*) ;

8. *Préparation du message de mise à jour :* si *update = true*, ce qui est toujours vrai pour $am_i = nw_i$, le nœud $i$ prépare un message de mise à jour $um_i$ pour tous ses voisins à un saut $j$, $j \in N_{1,i}$ avec

$$um_i \rightarrow * \ M, MAC_1, MAC_2, \dots, MAC_{\sigma_i}$$

Le message est construit de manière similaire au *neighborhood watch* message $nw_i$. Tous les messages $um_i$ pour tout $j \in N_{1,i}$ servent d'entrée à l'algorithme 2 et sont donc propagés vers le voisinage à un saut de $i$. Après le déclenchement de l'algorithme 2, l'indicateur de mise à jour est réinitialisé, c'est-à-dire *update = false*.

*Message reçu* $am_i = um_i$

Lorsque le nœud $i$ reçoit un message de mise à jour de liste de révocation $am_j = um_j$ de $j$, $i$ exécute les étapes 1-4 et 6-7 comme décrit dans le paragraphe précédent. Si le nœud $i$ met à jour au moins un de ses $k$-vecteurs, c'est-à-dire *update = true* et $hopcount > 1$, $i$ crée un message de mise à jour $um_i$ pour tout $j \in N_{1,i}$ avec $hopcount = hopcount - 1$, démarre l'algorithme 2 et réinitialise l'indicateur de mise à jour.

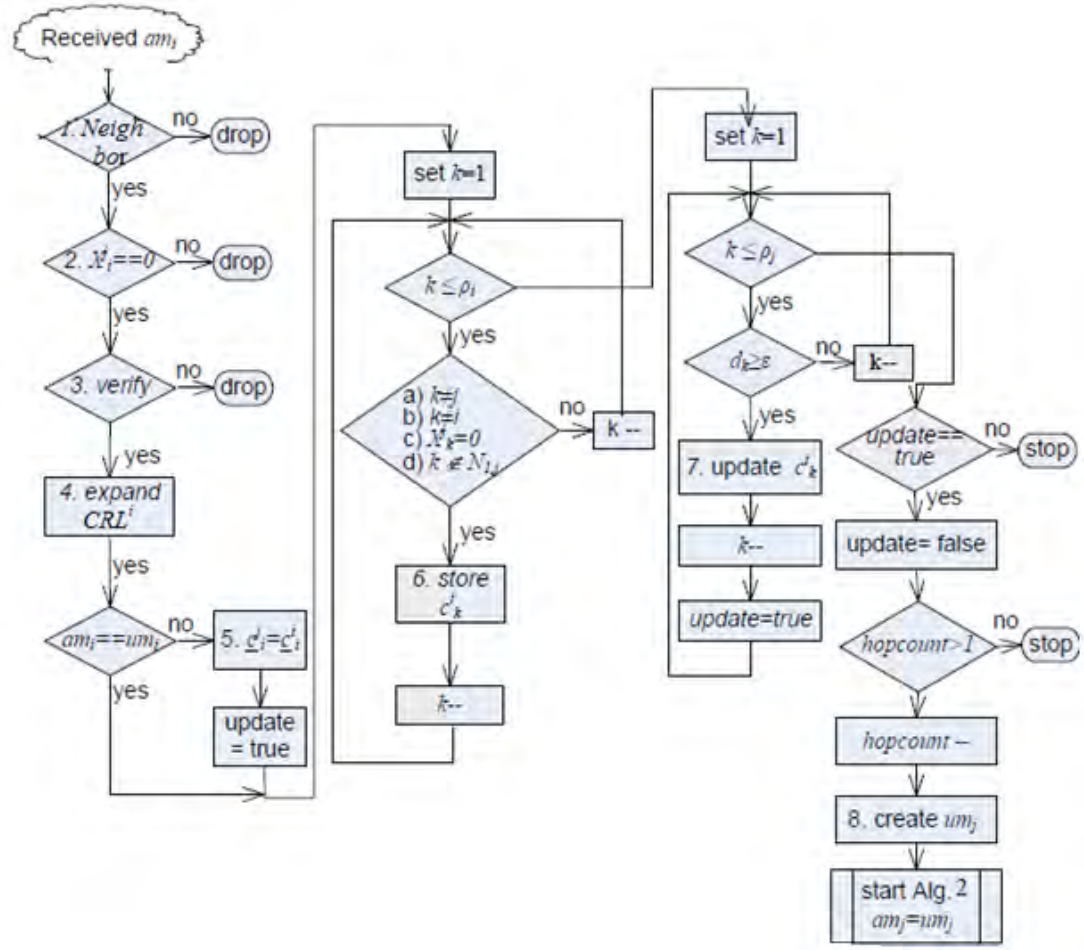



**Figure 6. : Organigramme de l'Algorithme *Update CRL***

## 6.4 Analyse de Sécurité et de Performance

### 6.4.1 Analyse de sécurité

La détection et la réponse aux attaques des adversaires membres du réseau sont des sujets des systèmes de détection d'intrusion et de réponse (IDRS). Cependant, pour garantir l'efficacité de tels systèmes, notre politique de sécurité doit être établie sur un protocole d'authentification sécurisée et efficace qui pourrait fournir un niveau élevé de protection contre les attaques de nœuds tiers externes au réseau. L'utilisation de HEAP comme protocole d'authentification fournit un fondement permettant de retirer efficacement du réseau un nœud compromis. Avec HEAP, SECRET assure l'authenticité et l'intégrité de chaque message au niveau de chaque saut. Par conséquent, il peut combattre diverses attaques provenant de nœuds tiers externes au réseau telles que :

- les attaques par usurpation d'identité ;
- les attaques DoS ;
- les attaques de type *man-in-the-middle* (MiM) ;
- Attaque du trou de ver (*wormhole*) ;
- les attaques par rejeu.

De plus, HEAP offre un certain niveau de protection contre les attaquants internes au réseau, qui essaient d'usurper d'autres nœuds internes au réseau en forgeant des messages.

Notre protocole de référence Hoeper et Gong (2009) présente un modèle de confiance paramétré, qui considère un large éventail d'attaques internes et propose l'utilisation de certaines techniques intelligentes, paramètres de sécurité, $\delta$ et $\varepsilon$, paramètres système $\alpha$ et $\beta$ et paramètre $m$ de propagation afin de les contrecarrer. Etant basé sur le même modèle de confiance paramétré SECRET parvient également à résister aux différentes attaques mentionnées dans l'analyse de sécurité fournie par [3] causées par les attaquants internes au réseau. Il s'agit notamment des :

- *Attaques de type Man-in-the-middle* : certains nœuds malveillants peuvent tenter d'usurper l'identité des autres nœuds ou de modifier le contenu des messages d'accusation, en particuliers ceux portés contre eux. Ceci est prévenu par le mécanisme d'authentification du broadcast utilisé, qui permet de vérifier au niveau de chaque saut l'authenticité et l'intégrité des messages reçus. Les nœuds malveillants peuvent également tenter de bloquer la transmission des accusations portées contre eux même. Ceci pourra être détecté par le système de surveillance du voisinage à un saut. De plus, puisque les accusations sont diffusées dans tout le voisinage elles pourront alors atteindre tous les autres nœuds du voisinage à $m$ sauts ;

- *Attaques sybil* : pour contourner le paramètre $\delta$ fixé comme seuil de révocation, un nœud attaquant peut fabriquer $\delta$ identités pour $\delta$ nœuds virtuels. Cependant, ce dernier ne sera pas en mesure de leur délivrer un certificat valide, puisque seule l'autorité de confiance *off-line* est habilitée à le faire après vérification de l'identité du nœud. Etant dans l'incapacité de fournir

aux nœuds virtuels un certificat et une clé publique valides, le nœud attaquant $j$ peut toujours tenter une autre forme d'attaque sybil en créant cette fois dans sa propre liste de révocation $\delta$ lignes pour $\delta$ nœuds virtuels $\in V = \{v_1, v_2, \dots, v_\delta\}$. Ceci afin de fixer $a_{i,x}^{j} = 1$ pour tout nœud $x$ dans $V$. Cependant la réussite d'une telle attaque demande la coalition d'au moins $\varepsilon$ nœuds, puisque les accusations rapportées reçues de $j$ ne sont acceptées que si au moins $(\varepsilon - 1)$ autres accusations rapportées sont reçues en provenance des mêmes nœuds dans $V$. Toujours dans le but de contourner le paramètre $\delta$, une autre forme d'attaque sybil consistant à répliquer les nœuds peut être lancée. Avec le mécanisme de révocation mis en place, les adversaires ne gagnent aucun avantage à procéder ainsi, puisqu'une seule accusation est comptée pour chaque nœud ;

- *Fausses accusations, attaques par coalition de nœuds, attaques par révocation de nœuds :* un adversaire peut tenter une attaque par révocation de nœud en modifiant sa propre liste de révocation de certificat. Cependant, l'impact de cette attaque est limité par les paramètres de sécurité $\delta$ et $\varepsilon$ :
    1. Les accusations directes d'un nœud malveillant comptent uniquement $1/\delta$ en raison de l'adoption du vote avec seuil lors du processus de révocation de clés ;
    2. $\varepsilon$ fausses accusations rapportées comptent uniquement $1/\delta$ en raison du seuil $\varepsilon$ fixé pour l'acceptation des accusations rapportées ;

- *Attaques par itinérance :* Un adversaire peut se déplacer à chaque fois que le nombre d'accusations portées contre lui approche $\delta$. Un adversaire peut aussi se déplacer vers un nouveau voisinage situé à plus de $m$ sauts, de sorte que personne n'ait une valeur d'accusation pour lui. Cependant dans les deux cas ses voisins actuels d'un saut détectent rapidement son comportement malveillant et révoquent localement son certificat. Avec les accusations se propageant vers tous les voisins à $m$ sauts, l'adversaire devra se déplacer plus rapidement que la propagation. Ainsi le pouvoir des adversaires itinérants est assez limité puisqu'ils doivent se déplacer fréquemment, et ne peuvent pas demeurer sur le même emplacement pour une longue période.

Le choix des paramètres de sécurité, $\delta$ et $\varepsilon$, des paramètres système $\alpha$ et $\beta$ et du paramètre de propagation $m$ pourra se faire conformément aux bornes et limites fixées dans [3], car dans celui-ci l'auteur :

- prend en compte les taux d'erreurs faux positifs (*false positive*) et faux négatifs (*false negative*) respectivement $\alpha$ et $\beta$, du système de surveillance implémenté au niveau des nœuds et montre

comment ils affectent la fonctionnalité et la sécurité des schémas de surveillance dans les MANETs ;

- montre comment les seuils de sécurité $\delta$ et $\varepsilon$ peuvent être utilisés pour contrecarrer les attaques sybil et les attaques par rejeu ;
- établit des limites et des relations pour les paramètres de sécurité $\delta$ et $\varepsilon$ et les paramètres du système de surveillance $\alpha$ et $\beta$ pour contrer les attaques des nœuds coalisés ;
- en outre, elle montre qu'en choisissant l'intervalle d'expiration des clés et le paramètre de propagation $m$ (portée de propagation des messages d'accusation) en concordance avec la vitesse de déplacement des adversaires itinérants, leur pouvoir pourrait être réduit. Elle souligne également que la menace posée par les adversaires mobiles peut être complètement contrecarrée en chiffrant toutes les accusations.

### 6.4.2 Analyse de Performance

En raison des contraintes en termes de ressources des MANETs et des exigences d'un mécanisme efficace de révocation de certificat, nous proposons d'évaluer la performance de notre proposition en nous intéressant à la charge de stockage, la charge de calcul, la charge de communication et le délai ou latence de la révocation.

Avec SECRET, pour participer au processus de révocation de clés d'un nœud compromis, chaque nœud doit stocker en plus de sa liste de révocation de certificats, $(\sigma_i + 1)$ clés, $\sigma_i$ clés publiques et $\sigma_i$ certificats.

Pour permettre l'authentification d'un *Neighborhood Watch* message $nw_i$ ou d'un message de mise à jour $um_i$, le nœud $i$ doit calculer un MAC en utilisant la clé de groupe $ikey$ et $\sigma_i$ MACs en utilisant les différentes clés de paire $okey_j$. Chaque nœud $j$ recevant ce message d'accusation calculera un MAC en utilisant sa clé $okey_j$. Ce qui engendre un coût de calcul égal à $(2\sigma_i + 1)$ MACs.

Un *Neighborhood Watch* message $nw_i$ ou un message de mise à jour $um_i$ contient $\sigma_i$ valeurs de hachage, une liste de révocation de certificat $CRL^i$, le numéro de série du certificat du nœud expéditeur $cert_i$, et un nombre $hopcount$ indiquant le nombre de sauts déjà accomplis. Ce message est transmis une seule fois par broadcast. Soit $l_M$ la longueur en termes d'octets d'un message d'accusation sans les valeurs de hachage représentant les MACs et $T_{hash}$ le temps requis pour hacher un octet de donnée d'entrée. Le délai $T_{auth}$ associé au calcul de valeurs d'authentification est alors égal à :

$$T_{auth} = (2\sigma_i + 1).\, l_M.\, T_{hash} \qquad (6.6)$$

Puisqu'avec notre solution, le message est transféré par broadcast vers le voisinage à un saut, nous obtenons alors pour le transfert un délai $T_{broadcast}$ égal à :

$$T_{broadcast} = (l_M + \sigma_i . l_{Mac}) . T_{transmit} \qquad (6.7)$$

$l_{Mac}$ étant la longueur en termes d'octets du digeste généré par le hachage du message et $T_{transmit}$ le délai requis pour la transmission et la réception d'un octet de donnée d'un saut vers un autre.

Pour évaluer les résultats issus de notre analyse de performance, nous évaluons également pour le protocole Hoeper et Gong (2009) le coût associé à la propagation d'un message d'accusation d'un saut vers un autre, en nous intéressant à la charge de stockage, la charge de calcul, la charge de communication et le délai de révocation.

Dans [28], pour participer à un processus de révocation de clé, un nœud a besoin de stocker sa liste de révocation de clés, $\sigma_i$ identités, $\sigma_i$ clés publiques et $\sigma_i$ clés de paire. Pour authentifier un *neighborhood Watch* message $nw_i$ ou un message de mise à jour de liste de révocation $um_i$, $2\sigma_i$ MACs doivent être calculés. Chaque message d'accusation inclut une valeur de hachage, une identité, une liste de révocation de clés et un nombre $hopcount$ déterminant le nombre de sauts accomplis. Puisque dans Hoeper et Gong (2009), le message d'accusation est transmis par unicast aux autres nœuds du voisinage, le nombre de messages engendré pour le voisinage à un saut égal à $\sigma_i$.

Le délai $T$ de propagation de ces messages devient donc égal à :

$$T = 2.\ \sigma_i\ .\ l_M . T_{hash} + \sigma_i\ .\ [(l_M + l_{Mac}) . T_{transmit}] \qquad (6.8)$$

Soient $ch_1$ et $ch_2$ les charges de calcul ou de communication respectives de Hoeper et Gong (2009) et de SECRET et $T_1$ et $T_2$ les délais respectifs de propagation des accusations vers le voisinage à un saut de Hoeper et Gong (2009) et de SECRET. Le tableau 6.3 donne les résultats de notre analyse comparative de performance.

| | **SECRET** | **Hoeper et Gong (2009) [28]** | **Comparaison** |
|---|---|---|---|
| **Charge de stockage** | – $\sigma_i$ *clés publiques* ;<br>– $(\sigma_i + 1)$ *clés* ;<br>– $\sigma_i$ *certificats* ;<br>– $CRL^i$ | – $\sigma_i$ *clés publiques* ;<br>– $\sigma_i$ *clés de paire* ;<br>– $\sigma_i$ *IDs* ;<br>– $KRL^i$ | $\boldsymbol{ch_1 < ch_2}$ |
| **Charge de calcul** | $(2\sigma_i + 1)$ *MACs* | $2\sigma_i$ *MACs* | $\boldsymbol{ch_1 \approx ch_2}$ |
| **Charge de Communication** | $\sigma_i(valeurs\ de\ hachage) +$ $1\ CRL + 1\ cert_i + 1\ hopcount$ | $\sigma_i(valeurs\ de\ hachage) +$ $\boldsymbol{\sigma_i}.[ID + KRL + hopcount]$ | $\boldsymbol{ch_1 \approx ch_2 + (\sigma_i - 1)\ .\ l_M}$ |

| | | | |
|---|---|---|---|
| **Délai de propagation de l'information** | $(2\sigma_i + 1).l_M.T_{hash} +$ $(l_M + \sigma_i.l_{Mac}).T_{transmit}$ | $2\sigma_i.l_M.T_{hash} +$ $\boldsymbol{\sigma_i}.(l_M + l_{Mac}).T_{transmit}$ | $\boldsymbol{T_1 \approx T_2 + [(\sigma_i - 1).l_M].T_{transmit}}$ |

**Tableau 6. : Comparaison de Performances**

L'évaluation comparative de performance de notre solution et celle proposée dans [28] dont les résultats sont donnés par le tableau 6.3 montre que :

- pour la sauvegarde d'informations requises pour l'authentification des messages et la vérification de la fiabilité des nœuds, SECRET nécessite plus d'espace de stockage que le protocole Hoeper et Gong (2009), donc $ch_1 < ch_2$. Ceci est principalement dû au stockage de certificats des nœuds voisins à la place de leur identités. Cependant, puisque l'approche proposée par HEAP peut être directement appliquée au protocole Hoeper et Gong (2009) pour l'authentification des échanges, le surcoût lié au stockage de certificats dans les systèmes basés sur PKI pourrait être évité ;
- les deux solutions engendrent presque la même charge de calcul, c'est-à-dire $ch_1 \approx ch_2$ vu que le surcoût de calcul lié à l'authentification d'un message d'accusation dans notre solution est négligeable puisqu'il équivaut juste au calcul d'un seul MAC supplémentaire (étape 1 dans HEAP) ;
- Avec SECRET, pour propager un message d'accusation vers son voisinage à un saut regroupant $\sigma_i$ noeuds, $i$ doit transférer une seule fois par broadcast un message contenant $\sigma_i$ valeurs de hachage, le numéro de série de son certificat, sa liste de révocation, et une valeur déterminant le nombre de sauts à accomplir. Par contre avec Hoeper et Gong (2009), pour propager la même accusation un message similaire contenant une valeur de hachage, l'identité de l'expéditeur, sa liste de révocation ainsi que le nombre de sauts à accomplir doit être envoyé par unicast pour couvrir tout le voisinage à un saut du nœud $i$, d'où pour le protocole Hoeper et Gong (2009) une charge de communication $ch_1$ presque égale à $ch_2 + (\sigma_i - 1).l_M$. Ainsi SECRET permet de réduire la charge de communication associée au protocole Hoeper et Gong (2009) de façon significative, soit de $(\sigma_i - 1).l_M$ octets dans le cas où l'identité d'un nœud et le numéro de série d'un certificat sont de même taille. Ce qui entrainera une réduction de l'occupation de la bande passante et de la consommation énergétique au niveau des nœuds participants. Ce gain se fera beaucoup plus sentir avec l'accroissement de la densité réseau et donc du nombre $\sigma_j$ de nœuds dans le voisinage d'un saut de chaque participant $j$ du réseau ;
- L'utilisation du transfert par broadcast réduit également le délai de propagation des accusations associé au protocole Hoeper et Gong (2009) d'une valeur égale à $[(\sigma_i - 1).l_M].T_{transmit}$. Ce qui engendre en termes de délai ou latence de la révocation un gain

significatif et ceci particulièrement avec l'accroissement de la densité du réseau, c'est-à-dire du nombre de nœuds voisins $\sigma_j$ de chaque participant $j$.

## Conclusion

Dans ce chapitre, nous avons adressé les insuffisances relatives aux protocoles de révocation distribués étudiés pour les MANETs en proposant un mécanisme de révocation de certificat qui adapte la proposition Hoeper et Gong (2009) aux systèmes basés sur PKI en utilisant HEAP comme protocole d'authentification du broadcast. Les analyses de sécurité et de performance présentées montrent que la combinaison de l'approche suggérée par HEAP et le modèle de gestion de la confiance de Hoeper nous permet d'obtenir :

- Une solution aussi robuste que le protocole Hoeper et Gong (2009), puisqu'elle combat efficacement les attaques de nœuds tiers externes au réseau, mais prévient ou résiste également aux attaques qui peuvent être lancées par les nœuds internes au réseau ;

- une solution qui améliore celle proposée par Hoeper et Gong (2009) en réduisant le coût de communication et le délai lié à la propagation des messages d'accusation ;

- une solution, qui, lorsqu'elle est appliquée à un système basé sur l'identité tel que Hoeper et Gong (2009), permettrait de réduire la charge de communication ainsi que le délai de propagation des messages de révocation sans générer le moindre surcoût en termes de stockage (élimination du stockage de certificats).

# Chapitre 7 SOLUTION POUR L'OPTIMISATION DU PROTOCOLE DE GESTION DE CLÉS "CL-EKM"

Dans ce chapitre nous présentons la solution que nous proposons dans le but d'adresser les insuffisances et limites soulignées dans le chapitre 5, relatives à la révocation de clés dans les WSNs. Cette présente contribution s'applique en général au domaine de la gestion des clés dans les WSNs, mais en particulier au domaine de la révocation de clés pour les systèmes basés sur une autorité centralisée.

## 7.1 Contexte et Motivation

Les résultats issus de notre analyse donnés dans le chapitre 5 montent que les principaux problèmes liés à la révocation de clés dans les WSNs résident dans le domaine de :

- l'efficacité ;
- la fiabilité ;
- la diffusion sécurisée de l'information de révocation par broadcast.

En effet, l'analyse des protocoles de révocation proposés pour les WSNs montre que chaque catégorie de solution présente des forces et des faiblesses. Plus spécifiquement, les solutions de la catégorie centralisée présentent un bon niveau de sécurité si et seulement si la ou les autorités impliquées sont hors de portée des adversaires. Toutefois avec ces protocoles, le transfert des messages de révocation par unicast affecte beaucoup leur efficacité car il implique une lourde charge de communication et un long délai de révocation surtout en cas d'accroissement de la taille du réseau (nombre d'éléments et/ou étendu) (voir tableau 7.1). Ce qui empêche la possibilité de passage à l'échelle de la presque la totalité des solutions de révocation centralisées.

**Tableau 7. : Performance des Protocoles centralisés – Synthèse de l'analyse**

| Protocoles Centralisés | Charge de Stockage | Charge de Communication | Charge de calcul | Délai de révocation |
|---|---|---|---|---|
| Eschnauer et Gligor (2002) [91] | $O(1)$ | $O(n)$ | $O(1)$ | $O(r+n)$ |
| Dini et Savino (2006) [92] | $O(log\ n)$ | $O(log\ n)$ | $O(1)$ | $O(r.log\ n)$ |
| Wang et al. (2007), Park et al. (2010) [93], [95] | $O(1)$ | --- | --- | --- |
| Mansour et al. (2014) [96] | $O(1)$ | $O(n)$ | $O(1)$ | $O(r+n)$ |
| Seo et al. (2015) [29] | $O(1)$ | $< O(n)$ | $O(1)$ | $< O(r+n)$ |

$n$ : nombre total de nœuds dans le réseau, $r$ : nombre total de sauts à parcourir.

Due à la coopération, les solutions de la catégorie distribuée, et généralement celles de la catégorie hybride, rencontrent à la fois des problèmes qui affectent leur sécurité et leur efficacité. Ceci parce que ces solutions sont vulnérables aux abus et engendrent également des charges de stockage, des charges calcul et des charges de communication pouvant épuiser les ressources limitées des capteurs. Ceci fait que les protocoles de ces deux catégories souffrent le plus des contraintes liées à la conception de solutions de révocation dans l'environnement des WSNs citées dans le chapitre 2 notamment la facilité de mise en œuvre des attaques et les contraintes en termes de ressources.

En raison des ressources limitées des capteurs, les protocoles proposés pour les WSNs en général et en particulier la plupart des protocoles adoptants l'approche distribuée sont basés sur la cryptographie à clé symétrique. Cependant, l'usage exclusif de la cryptographie symétrique pour assurer la protection des échanges complique d'avantage la situation, car ne garantit pas une résistance face à la compromission de nœuds, mais engendre plutôt des surcharges de communication et de stockage. Ce qui rend plus complexe le problème que pose la prise en charge des contraintes de conceptions mentionnées ci-dessus. Une solution pour répondre aux problèmes de sécurité rencontrés avec les protocoles de révocation adoptant l'approche distribuée consisterait à les renforcer par un mécanisme de gestion de la confiance comme c'est le cas avec les MANETs. Cependant, doter ces derniers d'un mécanisme de gestion de confiance afin de renforcer leur résistance face aux abus des nœuds participants affectera d'avantage leur efficacité en les rendant encore plus complexes et plus gourmands en ressources.

Pour toutes les raisons expliquées ci-dessus, qui se résument à la difficulté de fournir aux protocoles distribués et aux protocoles hybrides une ou des solutions qui adressent en même temps les problèmes liés à leur sécurité et à leur efficacité, nous décidons de nous focaliser sur les solutions de la catégorie centralisée. Plus spécifiquement nous nous concentrons sur le protocole centralisé montrant le plus d'atouts en termes de sécurité et d'efficacité notamment Seo et al. (2015) [29] et apportons des améliorations visant à adresser ses limites ou faiblesses exposées à la section II.1.

## 7.2 Fondements de notre Proposition

### 7.2.1 Choix du Protocole Seo et al. (2015) [29]

Dans le contexte des WSNs, la question liée à la gestion des clés a fait l'objet de nombreuses publications [4], [11-18], [124], [125]. Les contraintes en termes de ressources, poussent les concepteurs de solutions de gestion des clés à utiliser la cryptographie à clé symétrique. Cependant, plusieurs parmi leurs propositions souffrent de surcharge de communication et de mémoire, ne sont pas résistantes face à la compromission de nœuds et ne permettent pas le passage à l'échelle [29]. En outre, un protocole basé sur la cryptographie symétrique est incapable de prendre en charge la mobilité des nœuds, car celle-ci crée des complications qu'un tel protocole n'est pas en mesure d'adresser [74].

Pour ces dernières nous pouvons citer entre autres la détérioration de la qualité des liens de transmission établis, le changement fréquent de route, l'impossibilité pour le nœud mobile d'échanger avec de nouveaux voisins sans que ces derniers n'aient d'abord découvert sa présence et décider de la manière dont ils vont collaborer avec lui.

Pour surmonter les insuffisances liées aux protocoles de gestion des clés basés sur la cryptographie symétrique mentionnées ci-dessus notamment leurs problèmes de sécurité et leurs problèmes de performance, des approches basées sur des clés asymétriques ont été dernièrement proposées pour les WSNs dynamiques [126-129]. Bien que ces derniers efforts tirent parti de la cryptographie à clé publique (par exemple, la cryptographie à courbe elliptique (ECC)), la cryptographie à clé publique basée sur l'identité (ID-PKC), la résistance aux attaques de nœuds compromis, la flexibilité, l'extensibilité), ils montrent encore quelques faiblesses de sécurité [29].

Pour remédier aux insuffisances des propositions [126-129], Seo et al. ont proposé dans [29] un protocole connu sous le nom CL-EKM (*CertificateLess-Effective Key Management scheme*) qui aborde divers problèmes de sécurité [130], [131] et qui prend en compte l'aspect mobilité dans les réseaux de capteurs. Malgré les améliorations apportées par rapport aux propositions antérieures, CL-EKM présente certaines limitations critiques présentées dans [131], où les auteurs soulignent que CL-EKM souffre de coûts de calcul et de communication élevés, et ne répond pas adéquatement aux exigences d'extensibilité (scalabilité) de communication de groupe.

Nous remarquons que la plus grande partie de ces coûts globaux de calcul et de communication provient de l'utilisation intensive d'opérations de chiffrement et de monodiffusion (transfert par unicast) pour sécuriser les communications de la SB vers tous les nœuds chef de cluster (nœuds CH) du réseau. En fait, CL-EKM propose un mécanisme hiérarchique, basé sur le concept de clusters. Ainsi, pour effectuer la tâche de gestion des clés, ou la tâche de gestion des membres de cluster, la SB doit fréquemment communiquer avec tous les nœuds CH, afin de les informer des changements qui pourraient affecter le processus de gestion des clés. Pour assurer la protection de chaque communication, la SB doit chiffrer son message en utilisant la clé individuelle qu'elle partage avec chaque nœud CH avant de le transmettre par unicast. Pour des raisons de sécurité, les messages provenant principalement de la SB, et destinés à tous les nœuds CH, sont urgents et doivent atteindre les nœuds destinataires en temps opportun. Cependant, le chiffrement utilisé pour protéger les messages et le mode de transmission par unicast adopté peut entraîner de graves problèmes de performance (augmentation de la consommation d'énergie, de la bande passante et retard de propagation d'information) ; ceci spécialement quand :

*i)* une seule SB doit manager tout le réseau ;

*ii)* le réseau s'agrandit (nombre de composants et étendu) ;

*iii)* le nombre de messages à transmettre est important.

Sur la base de ces observations, nous jugeons tout comme les auteurs de [131] que le problème d'optimisation doit être pris en compte pour le protocole CL-EKM. Avec cette motivation, dans ce chapitre, nous proposons une approche pour améliorer son efficacité et son extensibilité (passage à l'échelle). Nous adoptons l'idée rapportée du travail dans [132] et présentons dans [133] le protocole *ECL-EKM (An Enhanced Certificateless Effective Key Management Protocol for Dynamic WSN)* qui évite l'utilisation intensive de chiffrement et de l'unicast pour sécuriser les communications entre la SB et tous les nœuds CH du réseau. Dans celui-ci, nous proposons un mécanisme d'authentification destiné à la diffusion par broadcast de messages qui :

1. réduit considérablement le coût associé aux communications entre la SB et tous les nœuds CH en termes d'énergie, de délai et de bande passante ;
2. améliore l'extensibilité de CL-EKM en permettant la livraison de messages de la SB à un grand nombre de nœuds CH dans un délai plus raisonnable et avec un coût de calcul et de communication plus acceptable.

### 7.2.2 Choix du Protocole d'authentification

Une solution évidente aux limitations mentionnées ci-dessus consiste à diffuser par broadcast les messages de la SB vers tous les nœuds CH afin de réduire le temps de diffusion et les ressources telles que l'énergie et la bande passante. Cependant, l'authentification de la diffusion par broadcast est un grand défi dans les WSNs avec de multiples problèmes non résolus tels que la charge de calcul, le retard dans le processus d'authentification, le besoin de synchronisation des horloges, les attaques DoS.

En cherchant une solution à ce problème, nous trouvons dans [132] une proposition qui aborde le problème d'authentification de la diffusion par broadcast dans les WSNs et fournit un moyen de surmonter les problèmes ouverts tels que ceux liés au délai d'authentification, au besoin de synchronisation, à la charge de calcul. En nous inspirant de ce travail nous proposons un mécanisme d'authentification de la diffusion par broadcast, qui dans des situations où plusieurs messages en attente sont à transférer de la SB aux différents nœuds CH du réseau, pourra efficacement remplacer le protocole d'authentification proposé par CL-EKM. Nous montrons alors que l'application de ce mécanisme d'authentification au protocole CL-EKM réduit considérablement le coût associé aux communications.

### 7.2.3 Vue d'ensemble de la solution dans [132]

Dans [132], un mécanisme d'authentification de la diffusion par broadcast basé sur PKC est proposé. Ce protocole évite l'utilisation intensive de signatures numériques à clé publique en exploitant une seule signature ECDSA (*Elliptic Curve Digital Signature Algorithm*) [134] pour l'authentification des $m$ messages dans $M = \{m_1, m_2, \dots, m_m\}$ diffusés par broadcast. Dans leur solution, les messages de $M$ à diffuser sont organisés en blocs étendus désignés par $EB_i, i = 0, \dots, K$. Chaque bloc étendu, sauf $EB_0$, contient $b$ messages lorsque $m$ est supposé être un multiple entier de $b$ ; c'est-à-dire, $m = Kb$. Si ce n'est pas le cas, le dernier bloc étendu $EB_K$ contiendra simplement ($m \bmod b$) messages, $K$ étant égal à $m / b$. Un bloc étendu est considéré comme une unité authentifiant un message à diffuser dans $M$. Le premier bloc étendu $EB_0$ contient un authentificateur qui est l'information supplémentaire (par exemple, le code d'authentification de message (MAC) ou la signature) pour authentifier un bloc étendu. En plus des $b$ messages de $M$ à diffuser, chaque bloc étendu, $EB_i, i = 1, \dots, K$ contient un authentificateur spécifié.

L'authentificateur dans le bloc étendu $EB_0$ est utilisé pour authentifier le bloc étendu $EB_1$ qui contient $b$ messages de $M$ à diffuser et un seul authentificateur. Pour se faire la valeur de hachage résistante aux collisions de $EB_1$ est calculée et comparée à l'authentificateur dans $EB_0$, s'il y'a égalité alors $EB_1$ est authentique. L'authentificateur de $EB_1$ est à son tour utilisé pour authentifier le bloc étendu $EB_2$ qui contient $b$ messages de $M$ à diffuser et un authentificateur. Le processus continue jusqu'au bloc étendu $EB_K$. Par conséquent, tous les messages de $M$ à diffuser peuvent être authentifiés en utilisant une seule signature.

Avant de décrire notre proposition dans la section suivante, dans le tableau 7.2, nous donnons la liste des principales notations utilisées dans ce chapitre ainsi que leur signification.

**Tableau 7. : Liste de Notations**

| | |
|---|---|
| $m$ | Nombre total de messages à transmettre à un instant donné de la SB vers tous les nœuds CH du réseau |
| $M$ | Vecteur $[m_i]\ i = 1, \dots, m$ au total $m$ messages à diffuser de la SB vers tous les nœuds CH du réseau |
| $N$ | Nombre total de nœuds CH dans le réseau |
| $r$ | Nombre total de sauts à parcourir pour atteindre tous les nœuds CH du réseau |
| $K$ | Nombre total de blocs étendus |
| $l_m$ | Taille d'un message (octets) |
| $l_d$ | Taille d'un digeste (octets) |
| $N_i$ | Nombre de nœuds CH situés à $i$ sauts par rapport à la SB |
| $E_r$ | Energie dépensée pour la réception de donnée de taille 1 octet |
| $E_e$ | Energie dépensée pour la transmission de donnée de taille 1 octet |
| $E_l$ | Energie dépensée pour écouter le canal (pendant une unité de temps) |

| | |
|---|---|
| $E_{dec}$ | Energie dépensée pour déchiffrer une donnée de taille 1octet |
| $E_{hash}$ | Energie dépensée pour le hachage d'une donnée de taille 1octet |
| $T_{ec}$ | Temps requis pour déchiffrer une donnée de taille 1octet |
| $T_{hash}$ | Temps requis pour calculer le digeste d'une donnée de taille 1octet |
| $T_{transmit}$ | Délai requis pour le transfert de donnée de taille 1 octet d'une distance égale à la portée de transmission des nœuds (d'un saut vers un autre) |
| $T_r$ | délai associé à la réception d'une donnée de taille 1octet |
| $T_e$ | délai associé à l'émission d'une donnée de taille 1octet |
| $k_{H_j}^0$ | clé individuelle partagée par la SB et le nœud CH $H_j$ |
| $E(k_{H_j}^0, m)$ | Message $m$ chiffré en utilisant la clé symétrique $k_{H_j}^0$ |

## 7.3 Description de ECL-EKM

Dans ECL-EKM, nous considérons que les messages sont transmis par broadcast de la SB à tous les nœuds CH du réseau. Par conséquent, nous considérons dans la suite uniquement le sous-réseau constitué de tous les nœuds CH dans le réseau. Ainsi, nous utilisons le terme broadcast au lieu de multicast dans la suite.

### 7.3.1 Vue d'ensemble de ECL-EKM

Afin de réduire le coût (énergie, délai) lié aux communications entre la SB et tous les nœuds CH dans CL-EKM, nous adoptons une approche similaire à celle de [132]. Dans notre solution, nous proposons de sécuriser l'authentificateur dans le bloc étendu $EB_0$ en le chiffrant à l'aide de la clé individuelle partagée entre la SB et chaque nœud CH avant de l'envoyer par unicast à chacun. Une fois cela fait, tous les $m$ messages de $M$pourront être envoyés par broadcast accompagnés des $K$ autres authentificateurs de la même manière que décrit dans [132].

### 7.3.2 Détails du Mécanisme d'authentification proposé par ECL-EKM

Comme dans [132], nous présentons notre approche en trois étapes :

1) génération de blocs étendus ;
2) diffusion par broadcast de blocs étendus ;
3) vérification de blocs étendus.

#### 7.3.2.1 Génération de blocs étendus

Pour générer les blocs étendus $EB$, les $m$ messages à diffuser sont partitionnés en $K$ blocs $B_1, B_2, \dots, B_K$. Chaque bloc $B_i, 1 \leq i \leq K$ contient $b$ des messages à diffuser ($| B_i | = b$) de sortes que le nombre total de messages à diffuser $m$ soit égal à :

$$m = Kb \qquad (7.1)$$

Pour générer chaque bloc étendu $EB_i$, les messages de $EB_i$ sont concaténés pour former une longue chaîne notée $CON\ (B_i)$ comme indiqué dans la formule (7.2).

$$CO\ (B_i)\ =\ m_{(i-1)b+1}||\ \dots\ ||m_{ib}\ , 1\ \leq\ i\ \leq\ K \qquad (7.2)$$

Alors, $PAD\ (CON\ (B_i))$ est obtenu en complétant $CON\ (B_i)$ avec l'authentificateur $d_{i+1}$ comme montré par la formule (7.3)

$$PAD\big(CON(B_i\ )\big) =\ CON(B_i)||d_{i+1}, 1\ \leq\ i\ \leq\ K \qquad (7.3)$$

ici, $d_{i+1}$ qui est le digeste de $PAD\ (CON\ (B_i))$ est à son tour calculé en hachant $PAD\ (CON\ (B_i))$ avec une fonction de hachage résistante aux collisions $H$ comme montré par la formule 7.4.

$$d_{i+1}\ =\ H(PAD(CON(B_{i+1})))\ , 1\ \leq\ i\ \leq\ K\ -\ 1 \qquad (7.4)$$

Notons que $CON\ (B_K)$ est complété avec $d_{K+1}$, qui est une chaîne de caractères aléatoire. Il convient également de noter que $EB_0$ ne contient qu'un authentificateur qui est le digeste $d_1$, c'est-à-dire la valeur de hachage résistante aux collisions de $PAD\ (CON\ (B_1))$, c'est-à-dire :

$$d_1\ =\ H\ (PAD\ (CON(B_1))) \qquad (7.5)$$

Comme mentionné dans la section précédente, nous utilisons ici la clé individuelle $k_{H_j}^0$ pour chiffrer $d_1$ du bloc étendu $EB_0$. Le bloc $B_i$ et le condensé $d_{i+1}$ comprennent le bloc étendu $EB_i$ c'est-à-dire :

$$EB_i\ =\ [B_i, d_{i+1}], 1\ \leq\ i\ \leq\ K \qquad (7.6)$$

L'algorithme 1 décrit ci-après donne la description complète de la génération de blocs étendus.

**Algorithme 1** Génération des blocs étendus $EB_i$

1. Partitionner les $m$ messages de $M$ à diffuser par broadcast en $K$ blocs $B_1, B_2, \dots\ , B_K$.
2. Initialisez $d_{K+1}$ avec une chaîne aléatoire de caractères.
3. Pour $i\ =\ K\ ;\ i\ \geq\ 1\ ;\ i\ =\ i\ -\ 1$ faire
4. Concaténer les messages dans $B_i$ pour générer $CON\ (B_i)$ comme indiqué par la formule (7.2)
5. Compléter $CON\ (Bi)$ avec le condensé ou digeste $d_{i+1}$ pour générer $PAD\ (CON(B_i))$ comme indiqué par la formule (7.3)
6. Calculer le condensé ou digeste $d_i$ de $PAD\ (CON(B_i))$ avec une fonction de hachage résistante aux collisions, comme indiqué par la formule (7.4)
7. $EB_i\ =\ [B_i,\ d_{i+1}]$

8. Fin pour
9. Chiffrer le digeste $d_1$ en utilisant la clé individuelle $k_{H_j}^0$ pour générer $EB_0 = E(k_{H_j}^0, d_1)$

La figure 7.1 ci-après illustre un exemple de génération de blocs étendus $EB$ pour neuf (9) messages à diffuser par broadcast. Dans celui-ci, trois (3) messages sont contenus dans chaque bloc.

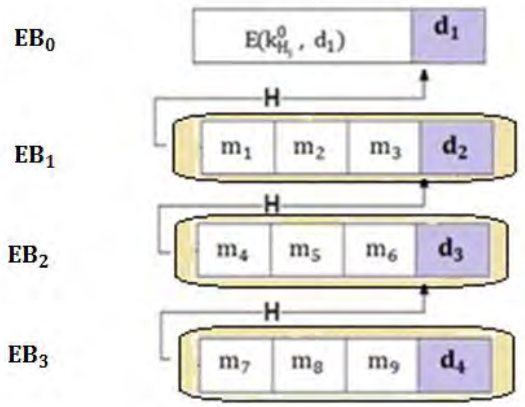



**Figure 7. : Un exemple de génération de blocs étendus**

#### 7.3.2.2 Diffusion de blocs étendus

Puisque l'authentification de $EB_i$ dépend de $EB_{i-1}$, cette dernière doit atteindre les nœuds CH récepteurs avant $EB_i$. Ceci est réalisé en utilisant le même mécanisme de diffusion séquentiel et fiable décrit dans [132]. Avec celui-ci, l'émetteur émet $EB_{i-1}$ avant $EB_i$. Les messages de chaque bloc étendu sont diffusés en fonction de leurs index, c'est-à-dire que la séquence d'envoi pour $EB_i$ est $m_{(i-1)\,b+1}, m_{(i-1)\,b+2}, \cdots, m_{ib}$. Le digeste $d_{i+1}$ dans $EB_i$ peut être envoyé avec un message de diffusion dans $EB_i$ car sa taille est relativement petite.

A la réception d'un message de diffusion $m_j$, le récepteur vérifie si $m_j$ appartient au bloc étendu actuel, par exemple $EB_i$, dont le digeste, $d_i$, a été reçu et authentifié avec $EB_{i-1}$. Si $m_j$ appartient à $EB_i$, le destinataire tente d'authentifier $EB_i$ et diffuse $m_j$ à ses voisins. Cela signifie que tous les destinataires échangent des messages d'$EB_i$ entre eux. Au cours de l'échange, un destinataire peut recevoir plusieurs copies de $m_j$ de différents voisins. Ainsi, le récepteur obtiendrait $m_j$ avec une probabilité élevée bien que chaque transmission ne soit pas fiable. En bref en faisant pleine utilisation de la transmission par diffusion, une transmission plus fiable est assurée. Si $m_j$ appartient au bloc $EB_{i+1}$ qui suit $EB_i$ et que $EB_i$ n'a pas encore été authentifié, cela signifie que tous les messages de $EB_i$ ont été diffusés. Par conséquent, le récepteur met en mémoire tampon $m_j$ et diffuse un accusé de

réception pour demander les messages manquants appartenant à $EB_i$. Après authentification de $EB_i$, $m_j$ sera diffusé.

#### 7.3.2.3 Vérification des blocs étendus

Selon l'étape de diffusion de bloc étendu, $EB_0$ atteint le nœud CH récepteur en premier. Ensuite, comme expliqué précédemment, $d_1$ contenu dans $EB_0$ est authentifié par chaque nœud récepteur grâce à sa clé individuelle $k_{H_j}^0$, qui permettra de déchiffrer $EB_0$. Une fois que $EB_0$ est authentifié, l'authenticité des autres blocs étendus, $EB_i$, $1 \leq i \leq K$ peut être vérifiée en utilisant une fonction de hachage résistante aux collisions. En effet, le digeste $d_i$, $1 \leq i \leq K$ dans $EB_{i-1}$ qui atteint les récepteurs à l'avance est utilisé pour authentifier $EB_i$. Donc, si :

$$H(m_{(i-1)b+1}|| \ \ldots \ ||m_{ib}||d_{i+1}) = d_i, \qquad (7.7)$$

$EB_i$ est authentique.

## 7.4 Analyse de Performance et Résultats

Dans cette section, nous évaluons et comparons la consommation énergétique et le délai associés aux communications entre la SB et les nœuds CH dans CL-EKM et ECL-EKM.

### 7.4.1 Consommation Energétique

#### 7.4.1.1 Evaluation de la consommation Energétique

Ici nous nous intéressons uniquement à l'énergie dépensée par l'ensemble des nœuds CH récepteurs des $m$ messages de $M$ en provenance de la SB. Ceci parce que cette dernière ne rencontre pas les mêmes contraintes en termes d'énergie que les nœuds du réseau.

Soient $E_{CL-EKM}$ et $E_{ECL-EKM}$ les énergies totales dépensées pour la transmission des $m$ messages de $M$ de la SB à tous les nœuds CH dans CL-EKM et dans ECL-EKM respectivement.

Dans CL-EKM, Pour recevoir un message de taille $l_m$, le nœud CH dépense une énergie d'écoute et de réception égale à $l_m . (E_l + E_r)$. Après réception de ce message, le nœud utilise sa clé individuelle $k_{H_j}^0$ pour le déchiffrer et vérifier son authenticité, ce qui consomme une énergie égale à $l_m . E_{dec}$.

Admettons que nous avons au total $r$ sauts dans le réseau et $N_i$ nœuds CH dans chaque saut $i$, $1 \leq i \leq r$. Puisqu'avec CL-EKM, chaque nœud CH reçoit $m$ messages transférés par unicast, nous avons à la réception au niveau de chaque saut :

$Saut\ 1 :$ $m\,.\,l_m.N_1.(E_l + E_r + E_{dec})$
$Saut\ 2 :$ $m\,.\,l_m.N_2.(2.E_l + 2.E_r + E_e + E_{dec})$
$Saut\ 3 :$ $m\,.\,l_m.N_3\,.(3.E_l + 3.E_r + 2.E_e + E_{dec})$
... ...
$Saut\ r :$ $m\,.\,l_m.N_r\,.(r.E_l + r.E_r + (r-1).E_e + E_{dec})$

Nous obtenons alors :

$$E_{L-EKM} = m\,.\,l_m.[E_{dec}\,.\sum_{i=1}^{r} N_i + (E_l + E_r).\ \sum_{i=1}^{r} iN_i\ +\ E_e\,.\sum_{i=1}^{r}(i-1).N_i] \qquad (7.8)$$

$$\boldsymbol{E_{CL-EKM} = m\,.\,l_m.[N\,.\,E_{dec}\ + (E_l + E_r).\ \sum_{i=1}^{r} iN_i\ +\ E_e\,.\sum_{i=1}^{r}(i-1).N_i]} \qquad (7.9)$$

Avec ECL-EKM, pour permettre l'authentification du premier message $EB_0$, chaque nœud CH doit exécuter une fois une opération de déchiffrement pour obtenir le digeste $d_1$ contenu dans $EB_0$.

Puisque $EB_0$ est transmis par unicast à l'ensemble des nœuds CH du réseau, la consommation énergétique $E_{ECL-EKM_1}$ générée par la réception et l'authentification de la source de ce message est de :

$$E_{ECL-EKM_1} = l_d.[N\,.\,E_{dec}\ + (E_l + E_r).\ \sum_{i=1}^{r} iN_i\ +\ E_e\,.\sum_{i=1}^{r}(i-1).N_i] \qquad (7.10)$$

Dans ECL-EKM, la réception des $m$ messages de $M$ et des $K$ digestes tous transmis par broadcast, engendre au niveau de chaque saut la consommation énergétique donnée ci-après :

$Saut\ 1 :$ $(m\,.\,l_m + K.l_d)\,.N_1.(E_l + E_r + E_e)$
$Saut\ 2 :$ $(m\,.\,l_m + K.l_d)\,.N_2.(E_l + E_r + E_e)$
$Saut\ 3 :$ $(m\,.\,l_m + K.l_d)\,.N_1.(E_l + E_r + E_e)$
... ...
$Saut\ r :$ $(m\,.\,l_m + K.l_d)\,.N_1.(E_l + E_r)$

Ce qui nous donne pour l'ensemble des nœuds CH du réseau, une consommation $E_{ECL-EKM_2}$ égale à :

$$E_{ECL-EKM_2} = (m\,.\,l_m + K.l_d).[(E_l + E_r)\,.\sum_{i=1}^{r} N_i + E_e\,.\sum_{i=1}^{r-1} N_i \qquad (7.11)$$

Etant donné que $m = Kb$, $(m\,.\,l_m + K.l_d)$ est égale à $K.L_{bloc}$, avec :

$$L_{bloc} = b\,.\,l_m + l_d \qquad (7.12)$$

nous obtenons alors :

$$E_{ECL-EKM_2} = K.L_{bloc}.[N.(E_l + E_r) + (N - N_r).E_e] \qquad (7.13)$$

Afin de vérifier l'authenticité et l'intégrité des messages, chaque nœud CH doit exécuter $K$ opérations de hachage pour générer $K$ digestes ou authentificateurs. Rappelons que $K$ est le nombre de blocs utilisés pour partitionner les $m$ messages de $M$ à diffuser. Ainsi $E_{ECL-EKM_3}$ est égal à :

$$E_{ECL-EKM_3} = N.K.\ L_{bloc}\ .E_{hash} \qquad (7.14)$$

Ce qui nous permet d'obtenir pour le protocole ECL-EKM une consommation énergétique globale égale à :

$$E_{CL-EKM} = E_{ECL-EKM_1} + E_{ECL-EKM_2} + E_{ECL-EKM_3} \qquad (7.15)$$

$$E_{ECL-EKM} = l_d.[N\ .\ E_{dec}\ + (E_l + E_r).\ \sum_{i=1}^{r} iN_i\ +\ E_e\ .\sum_{i=1}^{r}(i-1).N_i] + K.L_{bloc}.[N.(E_l + E_r) + (N - N_r).E_e] +\ N.K.L_{bloc}\ .E_{hash} \qquad (7.16)$$

$$\boldsymbol{E_{ECL-EKM} = N\ .[l_d\ .E_{dec} +\ K.\ L_{bloc}\ .E_{hash}] +\ l_d.[(E_l + E_r).\ \sum_{i=1}^{r} iN_i\ +\ E_e\ .\sum_{i=1}^{r}(i-1).N_i] +\ K.L_{bloc}.[N.(E_l + E_r) + (N - N_r).E_e]} \qquad (7.17)$$

#### 7.4.1.2 Comparaison de consommation énergétique

Pour quantifier les consommations énergétiques, nous considérons que chaque nœud CH est un SensorTag CC2650, qui est un appareil IoT (*Internet of Things*) développé par Texas Instruments [136] avec radio IEEE 802.15.4 intégrée. Il possède des capteurs de température, d'humidité, de pression, un magnétomètre, un gyroscope et un accéléromètre, ce qui en fait un outil puissant pour de nombreuses applications. Ce faisant, pour comparer les consommations énergétiques de CL-EKM et ECL-EKM, nous les quantifions en nous basant sur les valeurs données dans [135].

Pour les opérations cryptographiques, nous choisissons *AES* [63] comme algorithme de chiffrement et *Keccak* [137] comme fonction de hachage résistante aux collisions. Les coûts de communication sont composés des coûts d'écoute du canal et de réception, mais également des coûts d'émission des messages (voir Tableau 7.3). Nous supposons que chaque message à une longueur de 60 octets et que les digestes ou authentificateurs de blocs étendus ont une taille de 64 octets.

**Tableau 7. : Paramètres d'entrée (Energie)**

| Paramètres d'entrée | Valeurs |
|---|---|

| | |
|---|---|
| $l_m$ | 60 $bytes$ |
| $l_d$ | 64 $bytes$ |
| $E_l$+$E_r$ | 1,4 $\mu J$ |
| $E_e$ | 55,33$\mu J$ |
| $E_{dec}$ $(AES)$ | 0,021$\mu J$ |
| $E_{hash}$ $(Keccak)$ | 1,219$\mu J$ |

**Tableau 7. : Autres paramètres**

| Autres paramètres (lorsque fixés) | |
|---|---|
| $b$ | 3 |
| $m$ | 6 |
| $N$ | 160 |
| $r$ | 4 |

Les formules (7.9) et (7,17) montrent que les consommations énergétiques de CL-EKM et ECL-EKM dépendent des paramètres tels que la taille du réseau $N$, le nombre de sauts à parcourir $r$ et le nombre de messages $m$ à transmettre. Pour comparer ces consommations énergétiques et voir leur évolution en cas d'accroissement d'un ou de plusieurs de ces paramètres, nous les quantifions alors en :

1) fixant le nombre de messages $m$ de $M$ ainsi que le nombre de sauts $r$ à parcourir, et en faisant varier uniquement le nombre $N$ de nœuds CH dans le réseau. Le graphique de la figure 7.2 ci-dessous donne ce que nous obtenons comme résultat :

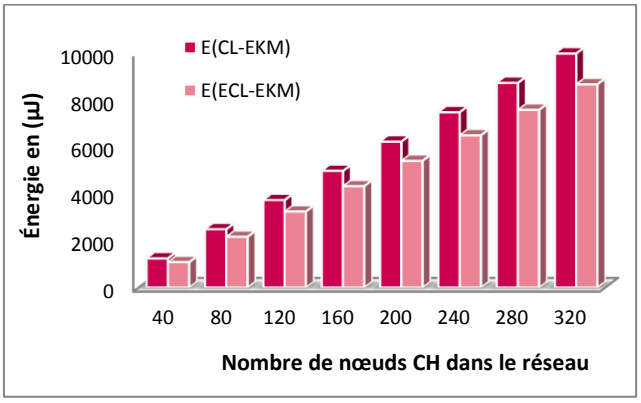



**Figure 7. : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de nœuds $N$)**

2) fixant nombre de messages $m$ de $M$, le nombre $N$ de nœuds CH dans le réseau et en faisant varier uniquement le nombre de sauts $r$ à parcourir, nous obtenons alors le graphique ci-dessous :

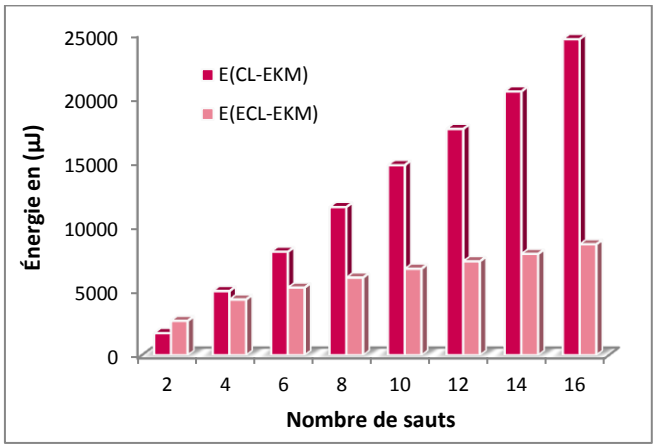



**Figure 7. : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de sauts $r$ à parcourir)**

3) en fixant uniquement le nombre de messages $m$ de $M$ et en faisant varier à la fois le nombre $N$ de nœuds CH et le nombre de sauts $r$ à parcourir, ce qui nous permet d'obtenir le graphique de la figure 7.4 ;

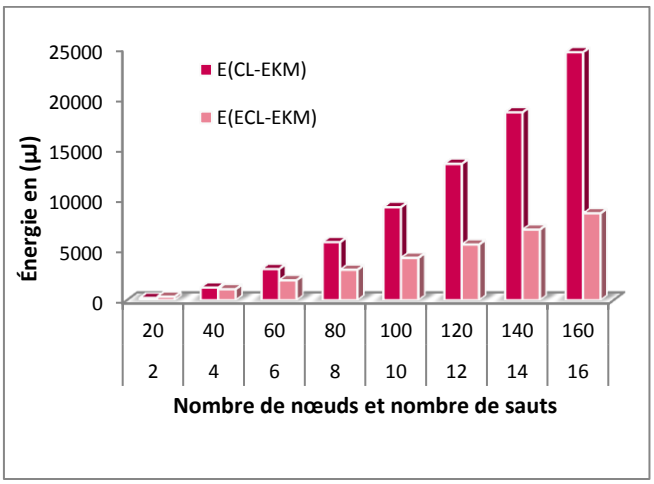



**Figure 7. : Evolution de la consommation énergétique en fonction de la variation de la taille du réseau (nombre de nœuds $N$ et nombre de sauts $r$ à parcourir)**

4) en ne faisant varier que le nombre de messages $m$ de $M$ à transmettre. La figure 7.5 ci-dessous montre ce que nous obtenons comme résultat.

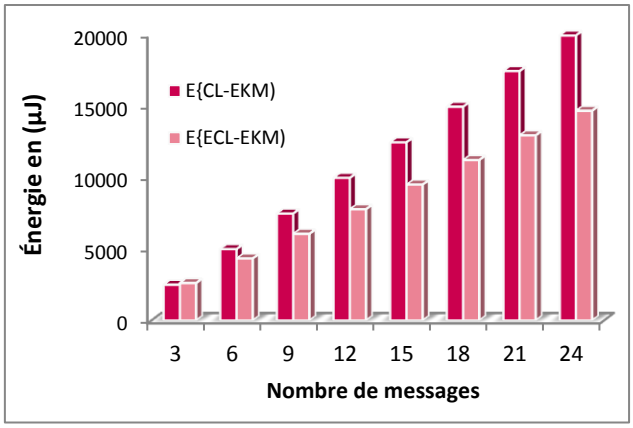



**Figure 7. : Evolution de la consommation énergétique en fonction de la variation du nombre de messages $m$ de $M$**

Les figures 7.2, 7.3, 7.4 et 7.5 montrent comme prévu que la consommation énergétique globale de CL-EKM croît très rapidement avec l'extension de la taille du réseau ou du nombre de messages à transmettre. Cette croissance rapide prend une allure exponentielle comme le montre la figure 7.3 lorsque le nombre d'éléments du réseau s'accroit ainsi que le nombre de sauts à parcourir.

Le protocole ECL-EKM réduit la consommation énergétique relative au protocole CL-EKM lorsque le réseau agrandit ou lorsque le nombre de messages de $M$ s'accroît. Le gain obtenu se fait beaucoup plus sentir avec les graphiques des figures 7.3 et 7.4. Pour ces deux scénarios, ceci se traduit par l'accroissement du nombre de retransmissions dans le cas du transfert des $m$ messages de $M$ par unicast vers tous les nœuds CH du réseau. En effet, puisque l'énergie requise pour l'émission d'une donnée de taille un octet a une valeur largement supérieure à toutes les autres (énergie d'écoute du canal et de réception, énergies liées aux opérations cryptographiques) (voir tableau 7.3), l'accroissement du nombre de sauts à parcourir accompagné ou non de l'accroissement du nombre de nœuds CH dans le réseau se traduit alors par l'accroissement de la consommation énergétique au niveau des nœuds relais.

ECL-EKM proposant une solution, qui évite l'utilisation intensive du transfert de messages de la SB vers tous les nœuds CH par unicast parvient alors à réduire la consommation énergétique de CL-EKM, ceci de façon significative lorsque le nombre de retransmissions s'accroît.

Dans nos calculs, nous avons considéré un réseau uniformément dense avec le même nombre de nœuds CH au niveau de chaque saut. Cependant, si nous sommes dans une situation où la densité est plus élevée au niveau des sauts les plus éloignés de la SB, qui est l'entité émettrice des messages, la réduction de la consommation énergétique observée avec les graphiques des figures 7.3 et 7.4 sera encore beaucoup plus importante.

### 7.4.2 Délai de Transfert

#### 7.4.2.1 Evaluation du Délai

Soient $T_{CL-EKM}$ et $T_{ECL-EKM}$ les délais respectifs de transfert des $m$ messages de $M$ dans les solutions CL-EKM et ECL-EKM.

Puisque dans CL-EKM, les messages transmis par unicast sont authentifiés grâce aux opérations de déchiffrement, nous avons au niveau de chaque saut :

$Saut\ 1 :$ $m\,.\,l_m\,.\,N_1.\,(T_{transmit} + T_r + T_{dec})$

$Saut\ 2 :$ $m\,.\,l_m\,.\,N_2.\,(2.T_{transmit} + 2.T_r + T_e + T_{dec})$

$Saut\ 3 :$ $m\,.\,l_m.\,N_3.\,(3.T_{transmit} + 3.T_r + 2.T_e + T_{dec})$

... ...

$Saut\ r :$ $m\,.\,l_m.\,N_r\,.\,(r.T_{transmit} + r.T_r + (r-1).T_e + T_{dec})$

Ce qui nous donne pour le transfert des $m$ messages de $M$ dans CL-EKM un délai global de :

$$\boldsymbol{T_{CL-EKM} = m\,.\,l_m.\,[N.T_{dec} + (T_{transmit} + T_r).\sum_{i=1}^{r} i.N_i + T_e\,.\sum_{i=1}^{r}(i-1).N_i]} \qquad (7.18)$$

D'après la formule (7.18), dans ECL-EKM, le délai $T_{ECL-EKM_1}$ lié au transfert par unicast du digeste $d_1$ contenu dans le bloc étendu $EB_0$ et à sa vérification est égal à :

$$T_{ECL-EKM_1} = l_d\,.\,[N.T_{dec} + (T_{transmit} + T_r).\sum_{i=1}^{r} i.N_i + T_e\,.\sum_{i=1}^{r}(i-1).N_i] \quad (7.19)$$

Avec le transfert par broadcast, au niveau de chaque saut les nœuds situés à la même distance par rapport au nœud émetteur reçoivent en même temps un message, et exécutent en même temps les vérifications. Le transfert des $m$ messages de $M$ et des $K$ digestes utilisés pour vérifier les $K$ blocs étendus engendre alors au niveau de chaque saut le délai suivant :

$Saut\ 1 :$ $K.\,l_{bloc}.\,(T_{transmit} + T_r + T_{hash} + T_e)$

$Saut\ 2 :$ $K.\,l_{bloc}\,.\,(T_{transmit} + T_r + T_{hash} + T_e)$

$Saut\ 3 :$ $K.\,l_{bloc}\,.\,(T_{transmit} + T_r + T_{hash} + T_e)$

... ...

$Saut\ r :$ $K.\,l_{boc}.\,(T_{transmit} + T_r + T_{hash})$

Ce qui nous permet d'obtenir pour l'ensemble des nœuds CH du réseau le délai $T_{ECL-EKM_2}$ égal à :

$$T_{ECL-EKM_2} = K.l_{bloc}.[r.(T_{transmit} + T_r + T_{hash}) + (r-1).T_e] \qquad (7.20)$$

Le délai global de ECL-EKM s'obtient alors en faisant la somme de $T_{ECL-EKM_1}$ et $T_{ECL-EKM_2}$. Ainsi, $T_{ECL-EKM}$ vaut :

$$\boldsymbol{T_{ECL-EKM} = l_d.[N.T_{dec} + (T_{transmit} + T_r).\sum_{i=1}^{r} i.N_i + T_e.\sum_{i=1}^{r}(i-1).N_i]}$$
$$\boldsymbol{+K.l_{bloc}.[r.(T_{transmit} + T_r + T_{hash}) + (r-1).T_e]} \qquad (7.21)$$

#### 7.4.2.2 Comparaison de Délais de transfert

Pour la quantification du délai, nous considérons le même type de nœuds CH que ceux décrits dans le paragraphe IV.1.2. Ce qui nous permet d'utiliser les valeurs données par [135] pour déduire les valeurs données dans le tableau 7,5.

A partir des formules (7.18) et (7.21) et des valeurs des tableaux 7.4 et 7.5, nous quantifions et comparons les délais associés au transfert des $m$ messages de $M$ des solutions CL-EKM et ECL-EKM :

**Tableau 7. : Paramètres d'entrée (Délai)**

| **Paramètres d'entrée** | **Valeurs** |
|---|---|
| $l_m$ | 60 $octets$ |
| $l_d$ | 64 $octets$ |
| $T_{transmit} + T_r$ | 1,98 $ms$ |
| $T_e$ | 1,86 $ms$ |
| $T_{dec}$ $(AES)$ | 0,002 $ms$ |
| $T_{hash}$ *(Keccak)* | 0,127 $ms$ |

Après variation des paramètres relatifs à la taille du réseau (nombre de nœuds et/ou nombre de sauts) et au nombre de messages, nous obtenons alors :

1) le graphique suivant en fixant le nombre de messages $m$ de $M$ ainsi que le nombre de sauts à parcourir $r$, et en faisant varier le nombre $N$de nœuds CH :

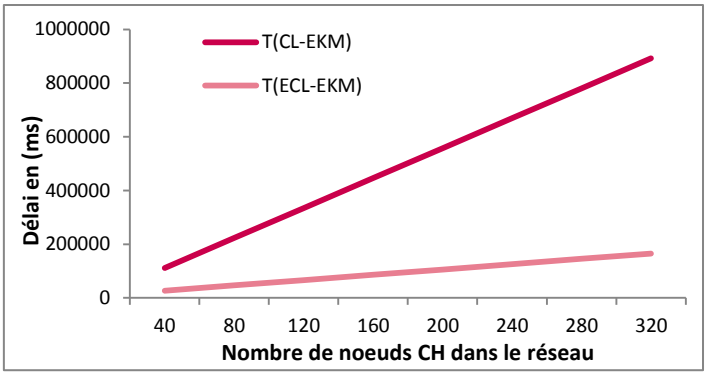



**Figure 7. : Délai de transmission en fonction de la variation de la taille du réseau (nombre de nœuds $N$)**

2) le graphique de la figure 7.7 ci-dessous en fixant le nombre de messages $m$ de $M$, le nombre de nœuds CH dans le réseau $N$et en faisant varier uniquement le nombre de sauts à parcourir $r$ ;

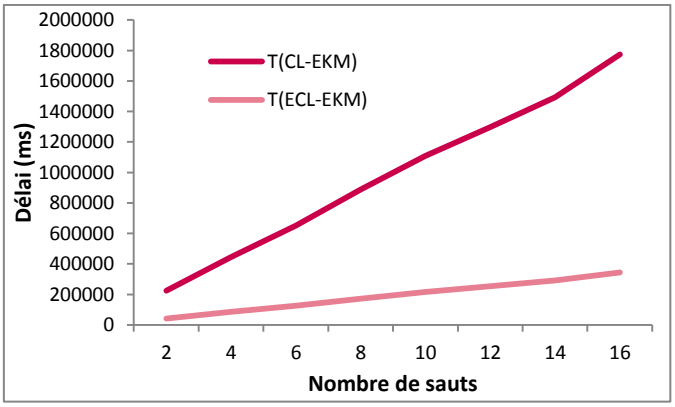



**Figure 7. : Délai de transmission en fonction de la variation de la taille du réseau (Nombre de sauts à parcourir $r$)**

3) le graphique ci-dessous en fixant uniquement le nombre de messages $m$ de $M$ et en faisant varier à la fois le nombre $N$ de nœuds CH et le nombre de sauts à parcourir $r$ ;

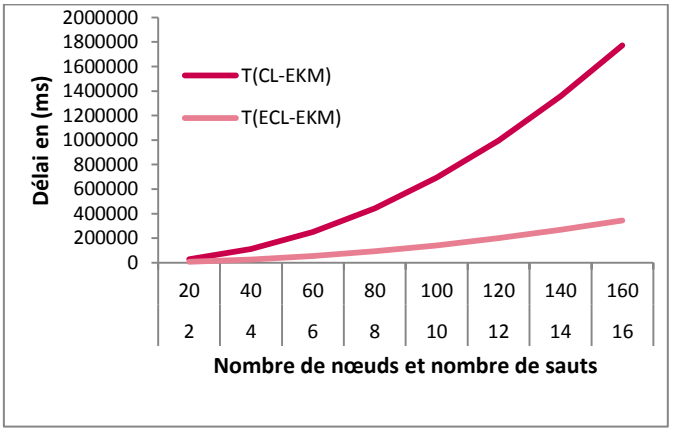



**Figure 7. : Délai de transmission en fonction de la variation de la taille du réseau**

**(Nombre de nœuds $N$ et nombre de sauts à parcourir $r$)**

4) le graphique de la figure 7.9 en ne faisant varier que le nombre de messages $m$ de $M$ à transmettre.

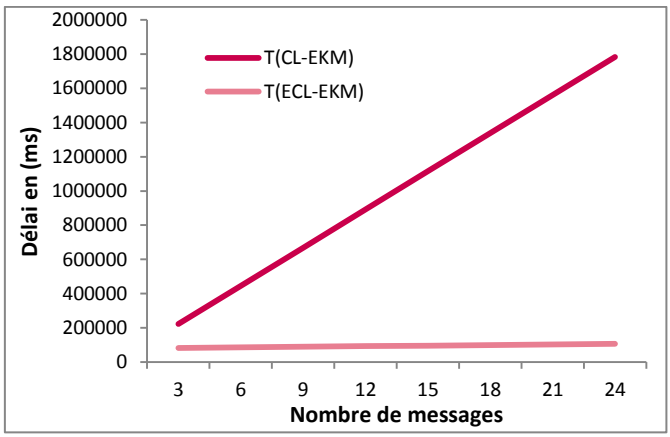



**Figure 7. : Délai de transmission en fonction de la variation du nombre de messages $m$ de $M$**

Les figures 7.6, 7.7, 7.8 et 7.9, permettent de voir qu'avec CL-EKM le délai s'accroît de la même manière que la consommation énergétique avec l'augmentation de la taille du réseau (nombre d'éléments et nombre de sauts) ou du nombre de messages $m$ à transférer.

Nous notons que le protocole ECL-EKM parvient à réduire considérablement le délai associé au transfert des messages dans CL-EKM et ce avec tous les scénarios testés c'est-à-dire quand la taille du réseau augmente (variation de $N$ et/ou de $r$) ou quand le nombre de messages $m$ s'accroît. L'explication à cela vient du fait que pour CL-EKM et pour ECL-EKM, l'énergie globale dépensée est égale à la somme des énergies dépensées individuellement par chaque nœud CH du réseau, et ce aussi bien pour le transfert par unicast que pour le transfert par broadcast. Il n'en est pas ainsi pour le délai de transfert des messages, car pour celui-ci, lorsque le mode de transmission par broadcast est utilisé comme c'est le cas avec ECL-EKM, les messages sont reçus en même temps par tous les nœuds situés à la même distance par rapport au nœud émetteur. Donc la durée de transfert ne sera comptabilisée qu'une seule fois pour tous les nœuds récepteurs indépendamment de leur nombre. C'est d'ailleurs pour cette raison que nous notons un plus grand écart entre les courbes du graphique de la figure 7.9. En effet, dans celui-ci, le délai associé à ECL-EKM semble se stabilisé, malgré l'accroissement du nombre de messages. Ceci parce que la taille du réseau ne varie pas avec ce scénario ni en nombre d'éléments ni en dimension, et donc la valeur du délai associé au transfert par unicast aussi utilisé dans ECL-EKM pour le transfert du digeste $d_1$ contenu dans $EB_0$ ne changera pas, d'où l'allure presque constante de la courbe représentant le délai de ECL-EKM.

## Conclusion

Dans ce chapitre, nous abordons le problème d'optimisation du protocole CL-EKM et présentons notre proposition ECL-EKM. Notre solution remplace le mécanisme d'authentification de CL-EKM par un mécanisme d'authentification du broadcast qui s'inspire de la proposition faite dans [132]. L'évaluation comparative fournie dans ce chapitre montre que ECL-EKM réduit la consommation énergétique et le délai associé au transfert des messages dans CL-EKM. Cette réduction se fait beaucoup plus sentir :

- Pour le délai lorsque la taille du réseau augmente en nombre d'éléments ou lorsque le nombre de messages à transmettre s'accroît ;
- Pour la consommation énergétique et le délai lorsque la taille du réseau augmente en nombre d'éléments et/ou en nombre de sauts ;

À partir de ces observations, nous confirmons que :

- Notre proposition améliore le protocole CL-EKM en réduisant la consommation énergétique et le délai de transfert des messages de gestion du réseau envoyés par la SB à tous les nœuds CH du réseau et par la même occasion rend le protocole de révocation de clés proposé par CL-EKM plus performant ;

- Notre proposition améliore le protocole CL-EKM en lui permettant un meilleur passage à l'échelle puisque améliore son efficacité, sa flexibilité et son évolutivité.

# CONCLUSION GENERALE

La gestion des clés est l'un des domaines de recherche les plus importants dans la sécurité des réseaux ad hoc en général et des MANETs et des WSNs en particulier. Celle-ci englobe deux aspects principaux que sont la distribution de clés et la révocation de clés en cas de compromission du nœud propriétaire.

Beaucoup de travaux ont été effectués dans le but d'avoir un schéma de gestion de clés performant qui assure un niveau élevé de sécurité tout en optimisant les métriques de performance et en conservant l'énergie. Cependant majoritairement ces travaux se sont plutôt orientés vers la distribution de clés aux nœuds du réseau et l'établissement de clés entre ces derniers en négligeant l'aspect fondamental qui est la révocation de clés. Pour pallier à ce déficit, des propositions de solutions de révocation de clés sont faites. Cependant en dépit des diverses approches et techniques utilisées par les concepteurs, les problèmes non résolus sont restés multiples dans l'environnement des MANETs en général et des WSNs en particulier. Cela nous a amené à nous investir dans cette thèse dans le but de fournir de manière exhaustive toutes les informations manquantes nécessaires pour une bonne compréhension du domaine et une meilleure prise en charge de la question.

A l'entame de nos travaux, nous nous étions proposé d'atteindre les objectifs suivants :

- Montrer l'importance de la révocation de clés dans le processus de sécurisation des MANETs et des WSNs via la gestion des clés ;

- Montrer le déséquilibre existant dans le domaine de la recherche quant à la gestion de clés dans les MANETs et les WSNs ;

- Etudier les différentes approches qui ont été adoptées pour la proposition de solutions de révocation de clés tant dans les MANETs que dans les WSNs ;

- Etudier et analyser les forces et faiblesses des approches et protocoles de révocation de clés dans les MANETs et dans les WSNs ;

- Rendre plus sécurisée et plus performantes les solutions de révocation de clés existantes pour les MANETs et les WSNs ;

- Donner de nouvelles orientations pour la résolution des problèmes de révocation de clés dans les MANETs et les WSNs.

## Synthèse

Le travail fourni est constitué de deux grandes parties. Pour atteindre les trois premiers sous objectifs susmentionnés, dans la première partie nous avons :

- Présenté les MANETs et des WSNs en mettant en relief leurs caractéristiques intrinsèques (absence d'infrastructure, topologie dynamique, mobilité, limitation en énergie etc.) et leurs vulnérabilités aux différentes attaques ;
- Montré le rôle primordial que joue la révocation de clés dans la fourniture de solutions de gestion de clés robustes capables d'adresser les menaces qui pourraient aboutir à la compromission de la confidentialité des clés, la compromission de l'authenticité des clés et l'usage non autorisé de ces dernières ;
- Fourni une classification des différents protocoles et donné la description détaillée sur le fonctionnement de chacun.

L'absence de travaux de recherche visant à évaluer et à comparer les solutions existantes exposées dans la première partie nous amène à proposer dans la deuxième partie de cette thèse une analyse des différents protocoles et catégories de solutions dans les MANETs et les WSNs. Cette analyse se base sur des critères de sécurité, de performance, mais également d'applicabilité. Ceci nous a permis d'atteindre notre quatrième sous objectif puisque nous avons pu :

1. Faire ressortir les avantages et inconvénients de chaque protocole, mais également les forces et faiblesses de chaque catégorie de solution.

2. Mettre la lumière sur les défis qui, à ce jour sont restés non résolus pour les différents protocoles et les différentes catégories de solutions.

Dans le but d'adresser les limites manquements soulignés lors de notre étude évaluative et ainsi atteindre le cinquième sous objectif que nous nous étions fixé, nous avons également décrit dans la deuxième partie des solutions qui adressent directement ou indirectement la question liée à la révocation de clés dans les MANETs et les WSNs.

Pour les MANETs, nous présentons alors notre protocole de révocation de certificat distribué nommé *SECRET* (*SECRET : A Secure and Efficient Certificate REvocation scheme for Mobile Ad hoc NeTworks*), qui est une adaptation de la solution qui s'est avérée comme présentant le plus d'atouts, notamment le protocole proposé dans [28] pour les systèmes à clés publiques basées sur l'identité. Cette adaptation est effectuée en vue de son application aux les systèmes à clés publiques basés sur le certificat. Afin de contourner les limites du protocole de référence, *SECRET* remplace le protocole

d'authentification fourni par [28] par le protocole *HEAP (Hop-by-hop Efficient Authentication Protocol)* décrit dans [119]. Ceci nous permet d'obtenir :

1. Pour les systèmes utilisant le certificat une solution de révocation aussi robuste que le protocole de référence [28] combattant les attaques potentielles venant de tiers internes ou externes au réseau ;

2. Un moyen d'améliorer la performance du protocole de référence [28] car l'application de notre proposition permettra également de réduire la charge de communication et le délai de propagation de l'information de révocation de celui-ci ;

3. Une solution qui pourra être réadaptée sans difficulté aux systèmes à clés publiques basées sur l'identité afin de résoudre les problèmes de performance rencontrés par les protocoles de révocation de clés suggérés pour ceux-ci.

Pour les WSNs, nous adressons indirectement les faiblesses liées à la solution de révocation montrant le plus d'atouts, notamment celle de la catégorie centralisée proposée par *CL-EKM* dans [29]. Pour ce faire nous avons présenté le protocole *ECL-EKM (An Enhanced Certificateless Effective Key Management Protocol for Dynamic WSN)*, qui utilise l'idée développée dans [132] afin de proposer un mécanisme d'authentification de la diffusion par broadcast de messages de la SB vers tous les nœuds CH du réseau. Ainsi *ECL-EKM* optimise *CL-EKM* en éliminant l'usage intensif du transfert par unicast de messages de la SB vers tous les nœuds CH.

Afin de vérifier la validité de cette proposition, nous avons évalué analytiquement les consommations énergétiques globales et délais associés à la propagation de messages des protocoles *ECL-EKM* et *CL-EKM*. Nous avons en outre comparé *ECL-EKM* et *CL-EKM* en étudiant leurs comportements en fonction de la taille du réseau ou du nombre de messages à transmettre. Pour cela nous nous sommes basés sur les valeurs expérimentales fournies par [135] pour quantifier les résultats issus de notre évaluation analytique, ce qui nous a permis de :

1. Confirmer l'accroissement rapide de la consommation énergétique et du délai de transmission de messages associé au protocole *CL-EKM* suite à l'accroissement de la taille du réseau (nombre d'éléments et/ou nombre de sauts) ou du nombre de messages lorsqu'une seule SB est impliquée ;

2. Montrer que *ECL-EKM* ne suit pas cette même tendance, puisqu'il optimise les performances de *CL-EKM* en réduisant sa consommation énergétique et son délai de transmission de messages.

# PERSPECTIVES

La conception de solutions de révocation de clés est une tâche très difficile et ceci plus particulièrement dans les WSNs parce qu'elle devra combiner les contraintes propres aux systèmes distribués et aux systèmes embarqués. Les perspectives ouvertes par ces travaux sont nombreuses et variées et les résultats issus de notre étude évaluative le confirment.

A l'issue de nos travaux nous dégageons les perspectives suivantes :

En ce qui concerne les MANETs, lors de la prochaine étape de validation de notre proposition, nous prévoyons :

1. De faire des simulations afin d'évaluer et comparer les charges de communication et délais de révocation de *SECRET* et du protocole de référence [28] afin de comparer nos résultats analytiques aux résultats expérimentaux obtenus ;

2. D'étudier le comportement des algorithmes en faisant varier les paramètres de sécurité $\delta$ (le seuil de révocation) et $\varepsilon$ (seuil pour le vote majoritaire) et le paramètre de performance $m$ (portée de propagation des votes) pour étudier le compromis performance / sécurité ;

En ce qui les WSNs, nous comptons également :

1. Evaluer et comparer par simulation les consommations énergétiques et délais de communication des protocoles *ECL-EKM* et *CL-EKM* dans le but de comparer nos résultats analytiques aux résultats expérimentaux obtenus ;

2. Etudier la possibilité d'améliorer les performances des protocoles de révocation centralisés en réduisant leur charge de communication et leur délai de révocation grâce au déploiement de plusieurs SBs au lieu d'une seule.

A l'issue de nos travaux, nous avons également noté plusieurs problèmes de recherche en suspens. Par conséquent pour atteindre le dernier sous objectif fixé, nous concluons en donnant les nouvelles orientations de recherche suivantes :

1. Dans les schémas de révocation distribués, chaque nœud doit pré-charger les votes contre tous les participants. Par conséquent, la topologie du réseau devra être connue avant le déploiement. Cela n'est pas toujours réalisable dans de nombreux déploiements réels. L'élimination de cette restriction rendra les systèmes distribués plus pratiques ;

2. Les protocoles de pré distribution basés sur le pool de clés manquent d'authentification de nœud à nœud et la révocation de nombreuses clés peut épuiser le pool de clés. Concevoir des protocoles de révocation efficaces et sécurisés est un défi. Il n'est donc pas surprenant qu'il existe peu, voire pas du tout, de travaux publiés sur ce sujet ;

3. Les schémas de cryptographie à clé publique ne semblent pas être le choix préféré dans les réseaux de capteurs, bien qu'ils puissent être plus appropriés que les schémas de cryptographie symétriques dans certains contextes de déploiement. Par exemple, l'utilisation appropriée de la cryptographie à clé publique peut entraîner un niveau de sécurité élevé à un coût abordable. Par conséquent, il est suggéré que des recherches supplémentaires puissent être entreprises pour déterminer la faisabilité du déploiement de systèmes basés sur la cryptographie à clé publique pour les WSNs ;

4. Toutes les solutions de révocation examinées dans le présent document nécessitent l'utilisation de systèmes de détection d'intrusion implémentés dans les nœuds. Des recherches supplémentaires pour améliorer l'exactitude et la rapidité de détection bénéficieront considérablement à la rapidité et à l'exactitude de révocation des nœuds compromis. Les techniques de défense contre les attaques courantes (attaques Sybil par exemple) doivent également être étudiées, en particulier dans les schémas de distribution de clés impliquant l'établissement de clé de chemin, afin de prévenir les attaques Sybil activées par clé de chemin ;

5. Il existe également un potentiel pour les solutions de renouvellement de clés non conçues à l'origine pour les WSNs, bien qu'un certain nombre de schémas existants proposés pour des réseaux ad hoc partageant des caractéristiques similaires (par exemple, MANETs et VANETs) ne soient pas adaptés au déploiement de WSNs. Ceci est principalement dû au fait que les WSNs contiennent des nœuds limités par les ressources et que, par conséquent, les performances des solutions destinées aux réseaux ad hoc partageant des caractéristiques similaires (par exemple, MANETs et VANETs) peuvent limiter leur adéquation au déploiement de WSNs. Un sujet de recherche potentiel est de proposer des schémas adaptés aux WSNs ;

6. Il est important de noter qu'en ce qui concerne les méthodes de révocation déjà proposées pour les MANETs et les WSNs, les nôtres incluses, il est essentiel de vérifier leur fonctionnalité dans un environnement simulé avant le déploiement dans un environnement de production, en particulier en tenant compte de la nature dynamique de ces réseaux.

## PUBLICATIONS

Mall, Dieynaba, Karim Konaté, and Al-Sakib Khan Pathan. On the Key Revocation Schemes in Wireless Sensor Networks. The 2013 IEEE International Conference on Green Computing and Communications (GreenCom 2013) : Security, Privacy, and Trust Computing, August 20-23, Beijing, China, 2013.

Mall, Dieynaba, Karim Konaté, and Al-Sakib Khan Pathan. SECRET : A Secure and Efficient Certificate Revocation Scheme for Mobile Ad Hoc Networks. International Symposium on Biometric and Security Technologies (ISBAST'14), August 26-27, 2014, Kuala Lumpur, Malaysia, 2014.

Mall, Dieynaba, Karim Konaté, and Al-Sakib Khan Pathan. Key Revocation in Wireless Sensor Networks – A Survey on a Less-addressed yet Vital Issue. International Journal of Ad Hoc and Ubiquitous Computing, Vol.18, No.1/2, pp.3 – 22, Inderscience Publishers, 2015.

Mall, Dieynaba, Karim Konaté, and Al-Sakib Khan Pathan. ECL-EKM : An Enhanced Certificateless Effective Key Management protocol for dynamic WSN. In *Networking, Systems and Security (NSysS), 2017 International Conference on*, pp. 150-155. IEEE, 2017.

# BIBLIOGRAPHIE


[1] Fokine K. Key Management in Ad Hoc Networks. *Master Thesis, Linkping University*, 2002. http://www.liu.se/.

[2] Capkun S., Buttyan L., and Hubaux J.-P. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, **2**(1), 52–64, 2003.

[3] Hoeper K. Authentication and Key Exchange in Mobile Ad Hoc Networks, *Ph.D. thesis*, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, 2007.

[4] Hegland A.M., Winjum E., Mjolsnes S.F., Rong C., Kure O., Spilling P. A survey of key management in ad hoc networks. *IEEE Communications Surveys & Tutorials*, Jul 1, 8(3), 48-66, 2006.

[5] Luo J., Hubaux J.-P., and Eugster P.T. DICTATE : Distributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks. *IEEE Transactionson Dependable and Secure Computing*, 2(4), 311–323, 2005.

[6] Luo H., Kong J., Zerfos P., Lu S., and Zhang L. URSA : Ubiquitous and Roubust Access Control for Mobile Ad Hoc Networks. *IEEE/ACM Transactions on Networking*, 12(6), 1049–1063, 2004

[7] Raya M. and Hubaux J.-P. Securing Vehicular Ad Hoc Networks, *Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks*, **15**(1), 39–68, 2007.

[8] Saxena N., Tsudik G., and Yi J.H. Identity-Based Access Control for Ad Hoc Groups, *Proceedings of the 7th International Conference on Information Security and Cryptology (ICISC 2004)*, ser. LNCS 3506, pp. 362–379, 2004.

[9] Yi S. and Kravets R. MOCA : Mobile Certificate Authority for Wireless Ad Hoc Networks, *Proceedings of the 2nd Annual PKI Research Workshop (PKI'03)*, pp. 65–79, 2003.

[10] Zhang Y., Liu W., Lou W., and Fang Y. Securing Mobile Ad Hoc Networks with Certificateless Public Keys, *IEEE Transactions on Dependable and Secure Computing*, **3**(4), 386–399, 2006.

[11] Zhou L. and Hass Z.J. Securing Ad Hoc Networks, *IEEE Networks Special Issue on Network Security*, 13(6), 24–30, 1999.

[12] Xiao Y., Ray V.K., Sun B., Du X., Hu F., Galloway M. A survey of key management schemes in wireless sensor networks. *Computer communications*. Sep 10,30(11-12) : 2314-41, 2007.

[13] Liu D., Ning P. Establishing pairwise keys in distributed sensor networks. *In : Proceedings of the 10th ACM conference on computer and communications security,Washington*, DC, USA, p.52–61,2003.

[14] Chen C.Y., Chao H.C. A survey of key distribution in wireless sensor networks. *Secur Commun Netw* 7(12):2495–508, 2014.

[15] Boyd C., Choo K-KR. Security of two-party identity-based key agreement. *In : Proceedings of progress in cryptology—Mycrypt*, p. 229–43, 2005.

[16] Choo K-K.R., Boyd C., Hitchcock Y. The importance of proofs of security for key establishment protocols: formal analysis of Jan-Chen,Yang-Shen-Shieh, Kim-Huh-Hwang-Lee,Lin-Sun-Hwang,andYeh-Sun protocols. *ComputCommun,* 29(15) : 2788–97, 2006.

[17] Ge M., Choo K-K.R., Wu H., Yu Y. Survey on key revocation mechanisms in wireless sensor networks. *Journal of Network and Computer Applications*, 63 : 24-38, Mar 1, 2016.

[18] Chan H., Gligor V. D., Perrig A., Muralidharan G. On the distribution and révocation of cryptographic keys in sensor networks. IEEE Trans Depend Secur Comput, 2(3) : 233–47, 2005.

[19] Wang Y., Ramamurthy B., Zou X. and Xue Y. An Efficient Scheme for Removing compromised Sensor Nodes from Wireless Sensor Networks. *CSE Technical reports.* Paper 77, 2007.

[20] Fan X., Gong G. A survey of key revocation schemes in mobile ad hoc networks. *In Handbook Of Security And Networks*,(pp. 265-281), 2011.

[21] Mall D., Konaté K. Analyse des protocoles de gestion des clés dans les réseaux Ad hoc, *Editions universitaires européennes*, 152 p, ISBN 978-613-1-59660-5, 2011

[22] Venkatraman L. and Agrawal D. P. A novel authentication scheme for ad hoc networks, *inProceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1268–1273, 2000.

[23] Candolin C. and Kari H. A security architecture for wireless ad hoc networks, in *Proceedings of IEEE Milcom 2002*, October 2002.

[24] Morogan M. C. and Muftic S. Certificate management in ad hoc networks, in *Symposium on Applications and the Internet Workshops (SAINT 2003)*, pp. 337–341, January 2003.

[25] Verma R. R. S., Mahony D., and Tewari H. Ntm - progressive trust negotiation in ad hoc networks, in *Proceedings of the 1st Joint IEI/IEE Symposium on Telecommunications Systems Research*, November 2001.

[26] Davis CR. A localized trust management scheme for ad hoc networks. *In Proceedings of the 3rd International Conference on Networking (ICN'04)* Mar 4 (pp. 671-675),2004.

[27] Moore T., Clulow J., Nagaraja S., Anderson R. New strategies for revocation in ad-hoc networks. *In European Workshop on Security in Ad-hoc and Sensor Networks* Jul 2 (pp. 232-246). Springer, Berlin, Heidelberg, 2007.

[28] Hoeper K., Gong G. Monitoring-based key revocation schemes for mobile ad hoc networks : Design and security analysis. *Technical Report 9 2009-15*, Centre for Applied Cryptographic Research, Mar 2009.

[29] Seo SH., Won J., Sultana S., Bertino E. Effective key management in dynamic wireless sensor networks. IEEE Transactions on Information Forensics and Security, 10(2) : 371-83, Feb 2015.

[30] Kupelian F. et Bollon P. Les réseaux sans-fils domestiques, Architecture Matérielle des Systèmes Informatiques, Rapport Février, pages 1-18, 2009.

[31] Atelin P. WiFi Réseaux sans fil 802.11 : Technologie - Déploiement –Sécurisation, Editions ENI, Rapport, pages 50, juillet 2010.

[32] Dioum B. Effets de la mobilité sur les protocoles de routage dans les réseaux ad hoc, thèse, Université Mouloud Mammeri de Tizi Ouzou (Algerie), pages 150, 2007.

[33] Gaye A. Proposition d'algorithmes sécurisés pour renforcer la coopération dans le routage des réseaux mobiles ad hoc, Thèse de doctorat, Université Cheikh Anta Diop de Dakar, 2016.

[34] Anjum F., Mouchtaris P. Security for wireless ad hoc networks, *Wiley*, 247 p, 2007.

[35] Sedrati M., Aouragh L., Guettala L., Bilami A. Etude des Performances des Protocoles de Routage dans les Réseaux Mobiles Ad-Hoc. *4th International Conference on Computer Integrated Manufacturing IP'2007*, pages 1-7, November 2007.

[36] Chanet J-P. Algorithme de routage coopératif à qualité de service pour des réseaux ad hoc agro-environnementaux, *thèse, No d'Ordre : 1745 EDSPIC : 373, Université Blaise Pascal - Clérmont II*, pages 158, Janvier 2009.

[37] Payal N. R., Prashant B. S. DPRAODV : A Dynamic Learning System against Blackhole Attack in Aodv Based Manet, *IJCSI International Journal of Computer Science Issues*, Vol. 2, Computer Engineering Department, SVMIT Bharuch, Gujarat, India, pages 1-6, September 2009.

[38] Ramaswamy S., Fu H., Sreekantaradhya M., Dixon J. and Nygard K. Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks. *Department of Computer Science, IACC 258 North Dakota State University, Fargo*, ND 58105, pages 1-7, Mars 2003.

[39] Emmanouil A. P., Levon N., Christos P. Securing AODV against Wormhole Attacks in Emergency MANET Multimedia Communications, *Wireless Multimedia & Networking (WMN) Research Group Kingston University London,* ICST 978-963-9799-62-2/00/0004, pages 1-7, July 2009.

[40] Yi P., Dai Z., Zhang S., Zhong Y. A New Routing Attack in Mobile Ad Hoc Network. *Department of Computing and Information Technology, Fudan University,Shanghai*, 200433, China, pages 1-12, June 2005

[41] MERAIHI R. Gestion de la qualité de service et contrôle de topologie dans les réseaux ad hoc, *thèse de doctorat, Ecole nationale supérieure des télécommunications*, France. janvier 2005.

[42] Khali S. Développement d'une technique de distribution de clés de cryptage pour les applications multicast sur les réseaux sans fil ad hoc. *Diss. (Mémoire de Maitrise)* Université du Québec à Trois-Rivières, 2008.

[43] Cordeiro M., Gossain H., and Agrawal D. P. MuIticast over wireless mobile ad hoc networks : present and future directions. *IEEE Network*, 17 : 52-59, Jan.-Feb. 2003.

[44] Akyildiz I. F., Su W., Sankarasubramaniam Y., and Cayirci E. A survey on sensor networks. *Communications Magazine, IEEE*, vol. 40, no 8, p. 102-114, 2002.

[45] Momani M., Agbinya J.I., Navarrete Guzman, G.P. and Akache M. A new algorithm of trust formation in wireless sensor networks. *In International conference on Wireless Broadband and Ultra Wideband Communication. UTS*, 2006.

[46] Zhao Q. and Tong L. Distributed opportunistic transmission for wireless sensor networks. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[47] Yick J., Mukherjee B., and Ghosal D. Wireless sensor network survey. *Computer networks*, vol. 52, no 12, p. 2292-2330, 2008.

[48] Ilyas M., and Mahgoub I. Handbook of Sensor Networks, *Compact Wireless and Wired Sensing Systems*, CRC PRESS 2004.

[49] Al-Karaki J.N. Infrastructureless wireless networks : cluster-based architectures and protocols, *PhD Thesis*, *Iowa State University*, 2004.

[50] Pathan A.-S. K.Security of Self-Organizing Networks : MANET, WSN, WMN, VANET., ISBN : 978-1-4398-1919-7, *Auerbach Publications, CRC Press, Taylor and Francis Group, USA*, 2010.

[51] Gayraud V., Nuaymi L., Dupont F., Gombault S., Tharon B. La sécurité dans les réseaux sans fil ad hoc. *In Symposium sur la Sécurité des Téchnologies de l'Information et de la Communication SSTIC*, *Rennes France*, June 2003.

[52] Yu X. A Defense System On DDos Attacks In Mobile Ad Hoc Networks, *Thesis Auburn University Alabama*, 2007.

[53] Simaremare H. A development of secure and optimized AODV routing protocol using ant algorithm, *Doctoral dissertation,Université de Haute Alsace*, 2013

[54] Marnerides A. Working with the Grid Kit Overlay Framework The Secure-AntHocNet Overlay, *Thesis Lancaster University*, 2007.

[55] Wood A. D. and Stankovic J. A. Denial of service in sensor networks, *IEEE Computer*, pp. 48-56, 2002.

[56] Labraoui N. La sécurité dans les réseaux sans fil ad hoc, *PhD Thesis, Université de Tlemcen*, 2012.

[57] Boyle D. and Newe T. Securing Wireless Sensor Networks : Security Architectures, *Journal of Networks*, ol. 3(1), 2008.

[58] Petit J. Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires, Thèse de doctorat, Université de Toulouse, 2011.

[59] Gong L. Increasing availability and security of an authentication service. *IEEE Journal on Selected Areas in Communications,* 11(5) : 657–662, June 1993

[60] Mehdi M., Anou A., Zair S., Bensebti M. and Djebari M. La Sécurité dans les Réseaux Ad Hoc, *In 4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2007,* March 2007.

[61] Ayachi M.A. Contributions à la détection des comportements malhonnêtes dans les réseaux ad hoc AODV par analyse de la confiance implicite, *Doctoral dissertation, Université Rennes 1 ; Université Européenne de Bretagne ; Université 7 Novembre à Carthage*, 2011.

[62] Kerckhoffs A. La cryptographie militaire. *Journal des sciences militaires*, IX : 5–83, 1883.

[63] Daemen J. and Rijmen V. The design of Rijndael : AES–the advanced encryption standard. *Springer*, 2002.

[64] Kahate A. Cryptography and network security. *Tata McGraw-Hill, 2 edition*, 2003.

[65] Rivest R.L., Shamir A., and Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) : 126, 1978.

[66] Krawczyk H., Bellare M., and Canetti R. HMAC : Keyed-Hashing for Message Authentication, http ://www.ietf.org/rfc/rfc2104.txt, Internet Draft, February 1997.

[67] Menezes A. J., van Oorschot P. C., Vanstone S. A. Handbook of Applied Cryptography, *CRC Press*, 816 p, 1996.

[68] Challal Y. Réseaux de capteurs sans fil. *University of Technology in compiegne, France*, 2008.

[69] Singh P. and Gill N. S. A Survey on Key management Schemes in Wireless Ad Hoc Networks. *International Journal of Applied Engineering Research*, *13*(1), pp.268-272, 2018.

[70] Balfe S., Yau P.W. and Paterson K.G. A guide to trust in mobile ad hoc networks. Security and Communication Networks, 3(6), pp.503-516, 2010.

[71] Yang H., Luo H., Ye F., Lu S.W. and Zhang L. Security in mobile ad hoc networks : challenges and solutions, *IEEE Wireless Communications* 11 (1) (2004) 38–47, 2004.

[72] Millen J.K. and Wright R.N. Certificate revocation the responsible way, *in Proceedings of Computer Security, Dependability and Assurance : From Needs to Solutions (CSDA'98), Washington,* D.C., , pp. 196–203, 1998.

[73] Liu W., Nishiyama H., Ansari N., NKato Y. J. Cluster-based certificate revocation with vindication capability for mobile ad hoc networks, In : IEEE transactions on parallel and distributed systems ,vol. 24, no. 2, p.239–49, 2013.

[74] Mall D., Konaté K. and Pathan A.S.K. Key revocation in wireless sensor networks – a survey on a less addressed yet vital issue, *International Journal of Ad Hoc and Ubiquitous Computing*, *18*(1-2), pp.3-22, 2015.

[75] Kong J., Zerfos P., Luo H., Lu S., and Zhang L. Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks, *Proceedings of the 9th International Conference on Network Protocols (ICNP'01)*, pp. 251–260, 2001.

[76] Crépeau C. and Davis C. R. A Certificate Revocation Scheme for Wireless Ad Hoc Networks, *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, pp. 54–61, 2003.

[77] Arboit G., Crépeau C., Davis C. R., and Maheswaran M. A Localized Certificate Revocation Scheme for Mobile Ad Hoc Networks, *Ad Hoc Network*, 6(1), 17–31, 2008.

[78] Hoeper K. and Gong G. Key Revocation for Identity-Based Schemes in Mobile Ad Hoc Networks, *Proceedings of the 5th International Conference on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW 2006), ser. LNCS* 4104, pp. 224–237, 2006.

[79] Fan X. and Gong G. Key Revocation Based on Dirichlet Multinomial Model for Mobile Ad Hoc Networks, *Centre for Applied Cryptographic Research (CACR) Technical Reports, CACR*, 2008.

[80] Gelman A., Carlin J.B., Stern H.S., and Rubin D.B. Bayesian Data Analysis, *Second Edition, Boca Raton, Florida, USA : Chapman & Hall/CRC*, 2004.

[81] Shafer G. A Mathematical Theory of Evidence, *Princeton University*, 1976.

[82] Chaib N., Lagraa N., Yagoubi M. and Lakas A. Unthresholded adaptive revocation technique in mobile ad hoc networks. *In Proceedings of the 8h ACM symposium on QoS and security for wireless and mobile networks* (pp. 75-78). ACM, October 2012.

[83] Guo C., Xu H., Ju L., Jia Z. and Xu J. A high-performance distributed certificate revocation scheme for mobile ad hoc networks. *In Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference* on (pp. 156-163). IEEE, September 2014.

[84] Clulow J, Moore T. Suicide for the common good : a new strategy for credential révocation in self-organizing systems. ACMSIGOPSOSR2006 ; 40(3) : 18–2.

[85] Park K., Nishiyama H., Ansari N., and Kato N. Certificate revocation to cope with false accusations in mobile ad hoc networks, *in Proc. 2010 IEEE 71st Vehicular Technology Conference : VTC2010-Spring, Taipei, Taiwan*, May 16-19, 2010.

[86] Liu W., Nishiyama H., Ansari N., and Kato N. A study on certificate revocation in mobile ad hoc network, *in IEEE International Conference 2011, Kyoto, Japan*, Jun. 2011.

[87] Kim S. Effective certificate revocation scheme based on weighted voting game approach. *IET Information Security*, *10*(4), pp.180-187, 2016.

[88] Izquierdo L.R., Izquierdo S.S. Dynamics of the Bush–Mosteller learning algorithm in 2 × 2 games, *InTech Publisher*, 2008.

[89] Kim S. Trust based dynamic bandwidth allocation scheme for Ethernet passive optical networks, *Wirel. Pers. Commun.*, 83, (4), pp. 2869–2882, 2015.

[90] Liu D, Ning P. Multi-level μTESLA : broadcast authentication for distributed sensor networks.Trans Embed Comput Syst ; 3(4) : 800–36, 2004.

[91] Eschenauer L., Gligor VD. A key-management scheme for distributed sensor networks, *In : ACM conference on computer and communication security*, p. 41–7, 2002.

[92] Dini G., Savino I. M. An efficient key revocation protocol for wireless sensor networks, *In : Proceedings of IEEE WOW-MOM'06*, p. 26–9, 2006.

[93] Wang Y., Ramamurthy B., Zou X. KeyRev : an efficient key revocation scheme for wireless sensor networks, *In : International conference on in communications*, p. 1260–5, 2007.

[94] Liu D., Ning P., and Sun K. Efficient self-healing group key distribution with revocation capability," in CCS '03 : Proceedings of the 10th ACM conference on Computer and communications security. New York, NY, USA : ACM Press, pp. 231–240, 2003.

[95] Park C-H., Zhang Y-Y., Kim I-T., Park M-S. DLS : dynamic level session key revocation protocol for wireless sensor networks. *In : 2010 International conference on information science and applications (ICISA)*, p. 1–8, 2010.

[96] Mansour I., Chalhoub G., Lafourcade P., Delobel F. Secure key renewal and revocation for wireless sensor networks, *In : 2014 IEEE 39th conference on local computer networks (LCN)*, p. 382–5, 2014.

[97] Seo S. and Bertino E. Elliptic curve cryptography based certificateless hybrid signcryption scheme without pairing, *CERIAS, West Lafayette, IN, USA, Tech. Rep. CERIAS TR 2013*-10, 2013.

[98] Chan H., Perrig A., Song D. Random key predistribution schemes for sensor networks, *In : IEEE symposium on security and privacy*, p. 197–213, 2003.

[99] Merkle R. Secure communication over insecure channels, *Communications of the ACM*, 21(4):294–299, 1978.

[100] Chao CH., Yang CF., Lin PT., Li JS. Novel Distributed key revocation scheme forwireless sensor networks, *Secur Commun Netw*, 6(10) : 1271–80, 2013.

[101] Blom R. An optimal class of symmetric key generation systems, *In : EUROCRYPT'84 on advances in cryptology*, p. 335–8, 1985.

[102] Chuang P-J., Chang S-H., Lin C-S. A Node Revocation scheme using public-key cryptography in wireless sensor networks. *J Inf Sci Eng,* 26(5) : 1859–73, 2010.

[103] Sánchez, D.S. and Baldus, H. Key management for mobile sensor networks. In *Proceedings of the First international conference on Secure Mobile Ad-hoc Networks and Sensors* (pp. 14-26). Springer-Verlag, September 2005.

[104] Hwang J., Kim Y. Revisiting random key pre-distribution schemes for wireless sensor networks, In : Proceedings of the 2nd ACM SASN workshop, p. 43–52, 2004.

[105] Pietro RD, Mancini LV., Mei, A., Panconesi A. Connectivity properties of secure wireless sensor networks, *In : Proceedings of the 2nd ACM SASN workshop*, p. 53–8, 2004.

[106] Chattopadhyay S., Turuk AK. A scheme for key revocation in wireless sensor networks*, Int J Adv Comput Eng Commun Technol*, 1(2) : 16–20, 2012.

[107] Jiang Y., Zhang RN., Du XJ. A new efficient random key revocation protocol for wireless sensor networks, *In : 14'th international conference on parallel and distributed computing, applications and technologies (PDCAT'13)*, 2013.

[108] Jiang Y., Shi H. A key pre-distribution scheme for wireless sensor networks using hexagonal deployment knowledge, *Chin J Electron*, 17(3) : 520–5, 2008.

[109] Ge M., Choo K-K R. A novel hybrid key revocation scheme for wireless sensor networks. In : Network and system security, p. 462–75, 2014.

[110] Ning, P., Liu, A. and Du, W. Mitigating DoS attacks against broadcast authentication in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *4*(1), p.1, 2008.

[111] Govindan, K. and Mohapatra, P. Trust computations and trust dynamics in mobile adhoc networks : A survey. *IEEE Communications Surveys & Tutorials*, *14* (2), pp. 279-298, 2012.

[112] Newsome J., Shi E., Song D., and Perrig A. The sybil attack in sensor networks : analysis & defenses, *in The 3rd international symposium on Information processing in sensor networks, IPSN '04*, pp. 259–268, 2004.

[113] Douceur J. R. The sybil attack, *inFirst International Workshop on Peer-to-Peer Systems, IPTPS* '01, pp. 251–260, 2002.

[114] Menezes A. J., Okamoto T., et Vanstone S. A. Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory*, vol. 39, no 5, p. 1639 -1646, sept. 1993.

[115] Sakai R., Ohgishi K., et Kasahara M. Cryptosystems based on pairing, *in SCIS 2000. Symposium on Cryptography and Information Security*, p. 26-28, 2000.

[116] Joux A. A One Round Protocol for Tripartite Diffie–Hellman, *in Algorithmic Number Theory*, vol. 1838, W. Bosma, Éd. Berlin, Heidelberg, p. 385-393, 2000.

[117] Moore T., Clulow J. Secure path-key revocation for symmetric key pre-distribution schemes in sensor networks, *In : International information security conference*, p. 157–68, 2007.

[118] Mall, D., Konaté, K. and Pathan, A.S.K. August. SECRET : A secure and efficient certificate revocation scheme for Mobile Ad hoc Networks. In *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on* (pp. 137-143). *IEEE*, 2014.

[119] Akbani R., Korkmaz T., and Raju G. V. S. HEAP : hop-by-hop efficient authentication protocol for Mobile Ad-hoc Networks, *In Proceedings of the 2007 spring simulation multiconference – Volume 1 (SpringSim '07), Vol. 1. Society for Computer Simulation International*, San Diego, CA, USA, pp. 157-165, 2007.

[120] Akbani, R. Defending against malicious nodes in closed MANETs through packet authentication and a hybrid trust management system, *PhD thesis, University of Texas at San Antonio*, 2009.

[121] Perrig A., Canetti R., Song D., and Tygar D. The TESLA Broadcast Authentication Protocol, *In RSA Cryptooctets*, pp. 2-13, 2002.

[122] Zhu S., Xu S., Setia S., and Jajodia S. LHAP : a lightweight hop-by-hop authentication protocol for ad-hoc networks, *23rd International Conference on Distributed Computing Systems Workshops*, pp. 749 – 755, 2003.

[123] Lu B. and Pooch U.W. A lightweight authentication protocol for mobile ad hoc networks, *International Conference on Information Technology : Coding and Computing, 2005 (ITCC 2005)*, Vol. 2, PP. 546 – 551, 2005.

[124] Gandino, F., Montrucchio, B. and Rebaudengo, M. Key management for static wireless sensor networks with node adding. *IEEE Transactions on Industrial Informatics*, 10(2), pp.1133-1143, 2014.

[125] Abdallah, W., Boudriga, N., Kim, D. and An, S. An efficient and scalable key management mechanism for wireless sensor networks. *In Advanced Communication Technology (ICACT), 2015 17th International Conference on July* (pp. 480-493). *IEEE*, 2015

[126] Rahman M. and El-Khatib K. Private key agreement and secure communication for heterogeneous sensor networks, *J. Parallel Distrib. Comput.*, vol. 70, no. 8, pp. 858–870, 2010.

[127] Alagheband M.R. and Aref M.R. Dynamic and secure key management model for hierarchical heterogeneous sensor networks, *IET Inf. Secur*., vol. 6, no. 4, pp. 271-280, 2012.

[128] Chatterjee K., De A., and Gupta D. An improved ID-based key management scheme in wireless sensor network, *in Proc. 3rd Int. Conf. ICSI*, vol. 7332, pp. 351–359, June 2012.

[129] Du D., Xiong H., and Wang H. An efficient key management scheme for wireless sensor networks, *Int. J. Distrib. Sensor Netw*., vol., Art. ID 406254, 2012.

[130] Kuchipudi R., Qyser A.A.M., and Balaram V.V.S.S.S. Latest Developments on Dynamic Key Management for Dynamic Wireless Sensor Networks, *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT 2016),* Chennai, India, 2016.

[131] Cheikhrouhou O. Secure Group Communication in Wireless Sensor Networks : A survey, *Journal of Network and Computer Applications*, Vol. 61, pp. 115-132, 2016.

[132] Liu Y., Li J., and Guizani M. PKC based broadcast authentication using signature amortization for WSNs, *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2106–2115, 2012.

[133] Mall, D., Konaté, K. and Pathan, A.S.K. ECL-EKM : An enhanced Certificateless Effective Key Management protocol for dynamic WSN. *In Networking, Systems and Security (NSysS)2017*, *International Conference on*(pp. 150-155),*IEEE*, January 2017.

[134] Johnson D., Menezes A. and Vanstone S. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security*, *1*(1), pp.36-63, 2001.

[135] Shibata, T., de Azevedo, R., Albertini, B.C. and Margi, C.B. Energy consumption and execution time characterization for the SensorTag IoT platform. *XXXIV Simpsio Brasileiro de Telecomunicaes e Processamento de Sinais (SBrT 2016)*, pp.55-59, 2016.

[136] Texas Instrument. Multi-standard cc2650 sensortag design guide, *TIDU862* – March 2015.

[137] Bertoni G., Daemen J., Peeters M., Van Assche G. The keccak reference, 2011.