ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3409

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire de Recherche en Informatique et Télécommunications
Discipline : Sciences de l'ingénieur
Spécialité : Informatique et Télécommunications

Présentée et soutenue le 25/12/2020 par :

Karim EL MARSSI

La sécurité des réseaux de capteurs sans fil et leur application pour la langue amazighe

JURY

Mohamed OUANNASSER	PES, Faculté des Sciences, Université Mohammed V - Rabat	Président
Mohamed EL MARRAKI	PES, Faculté des Sciences, Université Mohammed V - Rabat	Directeur de thèse
Hamid EZZAHRAOUI	PES, Faculté des Sciences, Université Mohammed V - Rabat	Rapporteur/Examinateur
Mohamed DAHCHOUR	PES, Institut national des postes et télécommunications - Rabat	Rapporteur/Examinateur
Lahoucine BALLIHI	PH, Faculté des Sciences, Université Mohammed V - Rabat	Rapporteur/Examinateur
Youssef EL AMRAOUI	PES, École nationale supérieure d'arts et métiers - Meknès	Examinateur
Ali KARTIT	PH, ENSA, Université Chouaib Doukkali - El Jadida	Co-encadrant

Année Universitaire : 2020-2021

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212 (05) 37 77 18 76 🖷 00212(05) 37 77 42 61; http://www. fsr.um5.ac.ma

AVANT-PROPOS

Les travaux présentés dans ce mémoire de thèse ont été effectués au Laboratoire de Recherche en Informatique et Télécommunications (LRIT), à la Faculté des Sciences de Rabat (FSR) Université Mohammed V - Rabat, Unité associée au CNRST (URAC29), sous la direction du professeur Mohamed EL MARRAKI, et l'co-encadrement de Ali KARTIT, professeur habilité à l'Université Chouaïb Doukkali d'El Jadida.

Je tiens à remercier vivement mon directeur de thèse, monsieur **Mohamed El MARRAKI**, professeur de l'enseignement supérieur à la faculté des sciences de Rabat pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Je suis également reconnaissant pour le temps qu'il m'a accordé, sa qualité pédagogique et scientifique, sa franchise et sa sympathie.

Je remercie aussi mon co-encadrant, monsieur **Ali KARTIT**, professeur habilité en Informatique, spécialité : Sécurité des Réseaux Informatiques à l'Université Chouaïb Doukkali d'El Jadida, pour l'aide et les conseils qu'il m'a apporté, notamment au début de la thèse.

Je remercie également monsieur **Mohamed OUANASSER**, professeur de l'enseignement supérieur à la faculté des sciences de Rabat pour sa disponibilité et d'avoir accepté de présider le jury de thèse.

Je remercie monsieur **Hamid EZZAHRAOUI**, professeur d'enseignement supérieur à la faculté des sciences de Rabat, d'avoir accepté sans hésitation de rapporter mon travail de thèse, et pour ses remarques constructives.

J'adresse également mes sincères remerciements à monsieur **Mohamed DAHCHOUR,** professeur d'enseignement supérieur à l'institut national des postes et télécommunications (INPT) de Rabat, pour sa disponibilité et pour avoir accepté d'être rapporteur de ma thèse.

Je remercie monsieur **Lahoucine BALLIHI**, professeur habilité à la faculté des sciences de Rabat, pour avoir accepté d'être rapporteur de ma thèse.

Je remercie monsieur **Youssef EL AMRAOUI**, professeur d'enseignement supérieur à la faculté des sciences de Rabat, et directeur de l'école nationale supérieure d'arts et métiers de Meknès (ENSAM), pour avoir accepté de consacrer son temps si précieux pour examiner ce manuscrit avec beaucoup d'intérêt.

En fin, merci à ceux que je n'ai pas pu citer, mais auxquels je réitère mes sincères Remerciements.

RÉSUMÉ

Les réseaux de capteurs sans fil ont gagné un intérêt majeur lors de la dernière décennie. Cependant, la sécurité reste un problème fondamentalement ouvert.

Notre mémoire s'articule autour de quatre contributions :

Le chapitre 3 examine les protocoles d'agrégation sécurisée de données proposés, à la pointe de la technologie. Ces protocoles peuvent être classés et analysés lors de l'agrégation de données chiffrées de saut par saut, bout en bout, et l'agrégation sécurisée de données non chiffrées. Dans le but d'évaluer l'efficacité des protocoles mentionnés.

Le chapitre 4 propose un schéma de signature sans certificat basé sur RSA, qui est un schéma largement appliqué dans des scénarios réels, en particulier pour les réseaux de capteurs sans fil. D'après les résultats, ce schéma est sécurisé dans le modèle d'oracles aléatoires et que sa sécurité est étroitement liée au problème de logarithme discret de RSA. En effet, nous avons prouvé qu'il était sûr sous certaines conditions contre les adversaires A_1 et A_2, ainsi il garantit également l'intégrité, la non répudiation et l'authentification.

Le chapitre 5 traite la reconnaissance des entités nommées REN des données des réseaux de capteurs sans fil. Cette approche permet aux informations structurées d'être interprétées sans ambiguïté. Une interprétation précise est une condition préalable nécessaire à la recherche, à la récupération et au traitement automatique des données des capteurs. Notre recherche sur RE a été appliqué sur la langue amazighe.

Le chapitre 6 propose une amélioration de la méthode de cartographie probabiliste de Koblitz en utilisant un message écrit en caractères Tifinaghe Unicode. L'amélioration proposée porte sur les variables de codage de l'algorithme de Koblitz pour faciliter la recherche des racines carrées de l'équation CE correspondant aux blocs du message *M*. Les résultats expérimentaux ont montré que notre algorithme de cartographie Koblitz modifié nécessite beaucoup moins de temps de codage qui ne varie pas nécessairement avec la taille du message *M*.

Mots-clés: Réseaux de capteurs sans fil, Sécurité, Cryptographie, Entités Nommés EN de la langue amazighe, Courbe elliptique CE, Protocoles de routage, Signature sans certificat.

ABSTRACT

Wireless sensor networks have gained major interest over the past decade. However, security remains a fundamentally open issue.

Our work revolves around four contributions:

Chapter 3 examines the proposed state-of-the-art secure data aggregation protocols. These protocols can be classified and analyzed during hop-by-hop, end-to-end, encrypted data aggregation and secure aggregation of unencrypted data. In order to evaluate the effectiveness of the mentioned protocols.

Chapter 4 provides an RSA-based certificateless signing scheme, which is a scheme widely applied in real-world scenarios, especially for wireless sensor networks. According to the results, this scheme is secure in the random oracle model and that its security is closely related to the discrete logarithm problem of RSA. Indeed, we have proven that it is safe under certain conditions against opponents A_1 and A_2, thus it also guarantees integrity, non-repudiation and authentication.

Chapter 5 deals with the recognition of entities named REN from data from wireless sensor networks. This approach allows structured information to be interpreted unambiguously. Accurate interpretation is a prerequisite for finding, retrieving and automatically processing sensor data. Our research on RE has been applied to the Amazigh language.

Chapter 6 proposes an improvement of the Koblitz probabilistic mapping method using a message written in Tifinaghe Unicode characters. The proposed improvement relates to the coding variables of the Koblitz algorithm to facilitate the search for the square roots of the CE equation corresponding to the blocks of the message M. The experimental results have shown that our modified Koblitz mapping algorithm requires much less encoding time which does not necessarily vary with the size of the M.

Keywords: Wireless sensor networks, Security, Named Entities of the Amazigh language, Elliptical curve, Cryptography, routing protocols, signature certificateless.

ⴰⵣⴳⵣⵍ (Résumé)

ⵉⵥⴹⵡⴰⵏ ⵏ ⵡⴰⵎⵎⴰⵥⵏ ⵡⴰⵔ ⵉⴼⵉⵍⵓ, ⵢⴰⵖⵓⵍ ⴷⴰⵔⵙ ⵢⴰⵏ ⵡⴰⵜⵉⴳ ⵎⵇⵇⵓⵔⵏ ⵉⵙⴳⴳⵯⴰⵙⵏ ⴰ ⵉⵣⵔⵉⵏ. ⵜⴰⵜⵉⴽⵏⵓⵍⵓⵊⵉⵜ ⴰⴷ ⵜⵜⵢⴰⵙⵙⵏ ⵓⴳⴰⵔ, ⵙ ⵓⵢⴰ ⴰⵙ ⵜⵔⵏⴰ ⴳ ⵓⴱⵓⵖⵍⵓ ⵏⵏⵙ, ⵀⵎⴰ ⴰⴷ ⵜⵔⴰⵔ ⴳⴼ ⵉⵙⵓⵜⵓⵔⵏ ⵉⴳⴳⵓⵜⵏ. ⴰⵙⵙ ⴰ, ⵉⵥⴹⵡⴰⵏ ⵏ ⵡⴰⵎⵎⴰⵥⵉ ⵜⵜⵓⵙⵙⵎⵔⵙⵏ ⴳ ⵜⴷⵓⵙⵉ ⴷ ⵜⴳⵯⴰⵎⵎⴰ ⴷ ⵓⵎⴰⵜⵔ ⴷ ⵜⴼⵍⵍⴰⵀⵜ ⴷ ⴰⴽⵯ ⴽⵉⴳⴰⵏ ⵏ ⵢⵉⴳⵔⴰⵏ ⵏ ⵜⵙⵏⵙⵉ. ⵎⴰⵛⴰ, ⴰⵏⴰⵡ ⵉ ⵉⵥⴹⵡⴰⵏ ⴰⴷ ⵉⵏⵏⵓⵍⴰ ⴰⵟⵟⴰⵚ ⵏ ⵜⵏⵥⵕⵉⵏ ⵏⵏⴰ ⵜⵓⴽⵣ ⵜⴳⵔⴰⵡⵜ ⵜⴰⵎⴰⵙⵙⴰⵏⵜ, ⵙ ⵓⵙⵜⵜⵉ ⵜⴰⴳⵔⴰⵡⵜ ⵏ ⵓⵣⵎⴰⵔ. ⵎⴰⴽⴰ ⵙⵙⵍⴰⵎⵜ ⴰⴷ ⵜⵙⵓⵍ ⵜⴳⴰ ⵜⴰⵎⵓⴽⵔⵉⵙⵜ ⵜⴰⴼⴳⴳⴰⴳⵜ.

ⴰⵙⵙⵎⵔⵙ ⵏ ⵉⵀⵓⴷⵓⵔⵏ ⵏ ⵓⵏⵏⴳⵣⵉ ⴰⴽⵔⵉⴱⵜⵓⴳⵔⴰⴼⵉⵜⵏ, ⵉⴳⴰ ⵢⴰⵏ ⵢⵉⴳⵔ ⵉⵜⵜⵎⵖⵓⵔⵏ, ⵎⴰⵛⴰ ⵉⵍⵍⴰ ⴰⵜⵉⴳ ⴳ ⵓⵔⵣⵣⵓ ⴰⵜⵔⴰⵔ ⵉⵥⵍⵉⵏ ⵙ ⵙⵙⵍⴰⵎⵜ ⵏ ⵓⵏⵖⵎⵉⵙ. ⵉⴷⴷⵖ ⵉⵀⵓⴷⵓⵔⵏ ⵏ ⵓⵏⵏⴳⵣⵉ ⴰⵔ ⴰⴽⴽⴰⵏ ⵉⴱⵖⵓⵔⵏ ⴳ ⵜⵉⵍⴰⵡⵜ ⵎⴳ ⵜⵏ ⵏⵙⵎⵢⵉⴼ ⴷ ⵉⵏⴳⵔⴰⵡⵏ ⵉⵇⵇⵏⵏ ⵙ ⵜⵙⴰⵔⵓⵜ ⵜⴰⴳⴷⵓⴷⴰⵏⵜ, ⴳ ⵜⵙⴳⴰ ⵏ ⵜⵉⴷⴷⵉ ⵏ ⵜⵙⴰⵔⵓⵜ ⴷ ⵓⵣⵎⵣ ⵏ ⵓⵙⵎⴽⵍ ⴷ ⵉⵥⵕⵉⵕⵉ ⵉⵣⵔⵔⵉⵏ. ⵉⵀⵓⴷⵓⵔⵏ ⵉⵏⵏⴳⵣⴰⵏ ⴰⵔ ⵣⵔⵔⵉⵏ ⵙ ⵓⵣⵔⴰⴱ ⴳ ⵓⵙⵙⵎⵔⵙ ⵏⵏⵙⵏ ⴳ ⵜⵙⵏⵙⵉⵡⵉⵏ ⴼⵙⵙⵓⵙⵏⵉⵏ.

ⵜⴰⵡⵓⵔⵉ ⵏⵏⵖ ⵜⵥⵍⵉ ⵙ ⴽⴽⵓⵥⵜ ⵏ ⵜⴱⵣⴰⵔⵉⵏ.

ⴰⴳⴹⴰⵎ ⵡⵉⵙⵙ ⴽⵕⴰⴹ ⴷⴰ ⵉⵇⵇⴰⵙ ⵉⴱⵕⵓⵟⵓⴽⵓⵍⵏ ⵏ ⵓⵙⵎⵓⵜⵜⵔ ⵉⵍⴰⵏ ⴰⴼⵔⴰⴳ ⵏ ⵜⵎⵓⵛⴰ ⵉⵜⵜⵓⴼⴽⴰⵏ ⵎⵉ ⴷⴰⵔ ⴰⵏⵙⴰ ⴰⴳⵯⵏⵙ ⵏ ⵜⴰⵜⵉⴽⵏⵓⵍⵓⵊⵉⵜ. ⵉⴱⵕⵓⵟⵓⴽⵓⵍⵏ ⴰⴷ ⵥⴹⴰⵕⵏ ⴰⴷ ⵜⵜⵓⴱⴹⵓⵏ ⵜⵜⵓⵖⵙⵏ ⴳ ⵓⵙⵎⵓⵜⵜⵔ ⵏ ⵜⵎⵓⵛⴰ ⵙ ⵉⵣⵡⵉⵍⵏ ⵉⵎⵉⴽ ⵙ ⵉⵎⵉⴽ ⴷ ⵓⵙⵎⵓⵜⵜⵔ ⵉⵍⴰⵏ ⴰⴼⵔⴰⴳ ⵏ ⵜⵎⵓⵛⴰ ⵜⴰⵔ ⵉⵣⵡⵉⵍⵏ, ⴷ ⵓⵙⵏⵏⴷ ⴳⴼ ⵉⵎⵉⴽⴰⵏⵉⵣⵎⵏ ⵏ ⵓⵣⵓⴳⴳⵣ ⵏ ⵓⵙⵎⵓⵜⵜⵔ ⵉⵍⴰⵏ ⴰⴼⵔⴰⴳ ⵏ ⵜⵎⵓⵛⴰ. ⵀⵎⴰ ⴰⴷ ⵏⵙⵜⵍ ⵜⵉⵕⵡⵉ ⵏ ⵉⴱⵕⵓⵟⵓⴽⵓⵍⵏ ⵡⴼ ⵏⵙⴰⵡⵍ.

ⴰⴳⴹⴰⵎ ⵡⵉⵙⵙ ⴽⵓⵥⵜ ⴷⴰ ⵢⵉⴽⴽⵉ ⵢⴰⵜ ⵜⵡⵏⴰⵖⵜ ⵏ ⵓⵙⴳⴹⵎ ⴰⴱⵍⴰ ⴰⵙⵍⴽⵉⵏ ⵉⴱⴷⴷⴰⵏ ⴳⴼ ”ⵔⵙⴰ”, ⵏⵏⴰ ⵉⴳⴰⵏ ⵜⴰⵡⵏⴰⵖⵜ ⵉⵜⵜⵓⵙⵏⵙⵏ ⴳ ⵉⵙⵉⵏⴰⵕⵢⵓⵜⵏ ⵏ ⵜⵉⵍⴰⵡⵜ, ⵙ ⵓⵙⵜⵜⵉ ⴳ ⵉⵥⴹⵡⴰⵏ ⵏ ⵡⴰⵎⴰⵥ ⵡⴰⵔ ⵉⴼⵉⵍⵓ. ⵜⵉⵢⴰⴼⵓⵜⵉⵏ ⵏⵏⴰ ⵏⵓⴼⴰ ⴰⵔ ⵎⵎⴰⵍⵏⵜ ⵉⴷ ⵜⴰⵡⵏⵖⴰⵜ ⵉⵍⴰⵏ ⴰⴼⵔⴰⴳ ⴳ ⵡⴰⵏⴰⵡ ⵏ ⵜⵡⵡⵓⵔⴰ ⵓⵔ ⵉⵕⵚⵚⴰⵏ, ⴷ ⵓⴼⵔⴰⴳ ⵉⵥⵍⵉ ⴽⵉⴳⴰⵏ ⵙ ⵜⵎⵓⴽⵔⵉⵙⵜ ⵏ ⵓⵍⵓⴳⴰⵔⵉⵜⵎ ⵉⴼⴼⵔⵏ ⵏ “ⵓⵙⵙ” ⴷ “ⵔⵙⴰ”. ⵙⵜⵉⵜ, ⵏⵎⵍⴰ ⵉⵙ ⵉⵍⴰ ⴰⴼⵔⴰ ⴳ ⴽⵔⴰ ⵏ ⵜⴼⴰⴷⵉⵡⵉⵏ ⵉⵜⵜⵓⵖⵙⴰⵏ ⴰⵟⵟⴰⵚ ⵎⴳⴰⵍ ⵏ ⵉⴼⵏⵉⵡⵏ ⵏ ⵡⴰⵏⴰⵡ ⴰ 1 ⴷ ⴰ 2, ⵙ ⵎⴽⴰ ⴰⵔ ⵉⴼⵔⵔⴳ ⵉ ⵓⵙⵎⵓⵏ ⴷ ⵜⵉⵍⵉ ⵏ ⵜⵉⵍⴰⵡⵜ ⵏⵏⵙ.

ⴰⴳⴹⴰⵎ ⵡⵉⵙⵙ ⵙⵎⵎⵓⵙ ⵢⵉⵡⵉ ⴷ ⴳⴼ ⵢⵉⴽⵣ ⵏ ⵉⴼⵔⴷⵉⵙⵏ ⵎⵉ ⵏⵇⵇⴰⵔ “ⵔⵙⴰ” ⵏ ⵜⵎⵓⵛⴰ ⵏ ⵉⵥⴹⵡⴰⵏ ⵏ ⵡⴰⵎⴰⵥⵏ ⵡⴰⵔ ⵉⴼⵉⵍⵓ. ⵜⴰⵖⴰⵔⴰⵙⵜ ⴰⴷ ⴰⵔ ⵜⵉⴽⴽⵉ ⵉ ⵉⵏⵖⵎⵉⵙⵏ ⵉⵜⵜⵓⵙⵏⵉⵏ ⴰⴷ ⵜⵜⵢⴰⴽⵣⵏ ⴱⵍⴰ ⴰⴳⵓ. ⵉⴽⵣ ⵉⵣⴷⴷⵉⴳⵏ ⵉⴳⴰ ⵜⴰⴼⴰⴷⴰ ⵜⴰⵎⵣⵡⴰⵔⵓⵜ ⵉ ⵓⵔⵣⵣⵓ ⴷ ⵓⵙⵙⵓⵖⵍ ⴷ ⵓⵙⵏⵙⵉ ⴰⵟⵓⵎⴰⵜⵉⴽⵉ ⵏ ⵜⵎⵓⵛⴰ ⵏ ⵡⴰⵎⴰⵥⵏ. ⴰⵔⵣⵣⵓ ⵏⵏⵖ ⴳⴼ “ⵔⵓ” ⵉⵥⵍⵉ ⵙ ⵜⵓⵜⵍⴰⵢⵜ ⵜⴰⵎⴰⵣⵉⵖⵜ.

ⴰⴳⴹⴰⵎ ⵡⵉⵙⵙ ⵚⴹⵉⵚ ⵉⵔⴰ ⴰⴷ ⵉⵙⵙⵀⵍⵓ ⵜⴰⵖⴰⵔⴰⵙⵜ ⵏ ⵜⴽⴰⵕⴹⵓⴳⵔⴰⴼⵉⵜ ⵉⵖⵉⵢⵏ ⴽⵓⴱⵍⵉⵣ ⵙ ⵓⵙⵎⵔⵙ ⵏ ⵢⴰⵏ ⵓⵎⴷⵢⴰ ⵏ ⵜⴱⵔⴰⵜ ⵉⵜⵜⵢⴰⵔⴰⵏ ⵙ ⵉⵙⵔⴽⵉⵍⵏ ⵏ ⵜⴼⵉⵏⴰⵖ ⵓⵏⵉⴽⵓⴷ. ⴰⵙⵙⵀⵍⵓ ⵏⵏⴰ ⵉⴽⴽⵉⵏ ⵉⵎⴰⵙⵙⵏ ⵓⵔ ⵉⵔⵙⵙⴰⵏ ⵏ ⵓⴽⵓⴷⴰⵊ ⵏ ⴰⵍⴳⵉⵓⵔⵉⵜⵎ ⵏ ⴽⵓⴱⵍⵉⵣ ⵀⵎⴰ ⴰⴷ ⵉⴼⵙⵓⵙ ⵓⵔⵣⵣⵓ ⴳⴼ ⵉⵥⵓⵕⴰⵏ ⵏ ⵍⵉⵇⵡⴰⵜⵢⵓⵏ “ⵓⵛ”, ⵎⵙⴰⵙⴰⵏ ⴷ ⵜⴱⵔⴰⵜⵉⵏ “ⵎ”. ⵀⵎⴰ ⴰⴷ ⵏⵥⵉⵕ ⵜⵓⴳⴳⵉⵜ ⵉⵖⵓⴷⴰⵏ ⵏ ⵡⴰⵍⴳⵓⵕⵉⵜⵎ ⵏⵙⵎⵢⴰⴼ ⵏⴳⵔ ⵏ ⵜⵖⴰⵔⴰⵙⵉⵏ: ⵜⴰⵎⵣⵡⴰⵔⵓⵜ ⵜⵟⵟⴼ ⵙ ⵢⴰⵜ ⵏⵇⵇⵉⴹⵜ ⵜⴰⵎⵙⵙⵉⵔⵉⵡⵜ “ⴳⵯ” ⵏ ⵢⴰⵏ ⵓⴳⵔⴰⵡ ⵎⵥⵥⵉⵢⵏ “ⵓⵛ” ⴷ ⵢⴰⵜ ⵜⵏⴰⵎⵎⴰⵙⵜ ⵉⵙⵎⵓⵜⵜⵓⵢⵏ ⴰⵏⵙⴰ ⵏ ⵜⵏⵇⵇⴰⴹ. ⵜⵉⵙⵙ ⵙⵏⴰⵜ ⵜⵙⵎⵔⵙ ⵢⴰⵏ ⵓⵛⵉⵎⴰ ⵏ ⵓⴽⵓⴹ “ⵓⵛⵛ” ⵉⴳⴰⵏ ⵙⵉⵏ ⵜⵇⵇⵏ “ⴰⴷⵏ”. ⵜⵉⵢⴰⴼⵓⵜⵉⵏ ⵏ ⵢⵉⵔⵉⵎ ⵏ ⵓⵙⵎⵢⴰⴼⴰ ⵎⵍⴰⵏⵜ ⵉⴷ ⴰⵍⴳⵓⵔⵉⵜⵎ ⵏⵏⵖ ⵏ ⵜⴽⴰⵕⴹⵓⴳⵕⴰⵜⵉⴼ ⴽⵓⴱⵍⵉⵣ ⵉⴱⴷⴷⵍⵏ, ⴰⵔ ⵉⵙⵙⵓⵜⵓⵔ ⴰⵣⵎⵣ ⵉⴳⵣⵣⵓⵍⵏ ⵏ ⴽⵓⴷⴰⵊ ⵓⵔ ⵉⵥⵍⵉⵏ ⵙ ⵜⵉⴷⴷⵉ ⵏ ⵜⴱⵔⴰⵜ “ ⵎ”.ⵜⵉⵢⴰⴼⵓⵜⵉⵏ ⴰⴷ ⵥⴹⴰⵕⵏⵜ ⴰⴷ ⵣⵣⵔⴱⵏⵜ ⵜⵉⴽⵍⵉ ⵏ ⵓⵎⴰⴱⴱⴰⵊ ⵙⵙⵔⴷⴰⵙⵏⵜ ⴰⵣⵎⴰⵔ ⵏ ⵉⴽⵔⵉⴱⵜⵓⵙⵢⵙⵜⴰⵎ ⵉⴱⴷⴷⴰⵏ ⴳⴼ “ⵓⵛ”.

ⵜⵉⴳⵓⵔⵉⵡⵉⵏ ⵏ ⵜⵓⵙⵓⵔⴰ: ⵉⵥⴹⵡⴰⵏ ⵏ ⵡⴰⵎⵎⴰⵥⵏ ⵡⴰⵔ ⵉⴼⵉⵍⵓ, ⴰⴼⵔⴰⴳ, ⵜⵍⵉⵜⵉⵏ ⵙ ⵡⴰⵙⵙⴰⵖ ⵏ ⵜⵓⵜⵍⴰⵢⵜ ⵜⴰⵎⴰⵣⵉⵖⵜ, ⴰⵀⵓⴷⵔ ⵉⵏⵏⴳⵣⴰⵏ, ⴰⴽⵔⵉⴱⵜⵓⴳⵔⴰⴼⵉ, ⴰⴱⵕⵓⵟⵓⴽⵓⵍ ⵏ ⵓⵙⵏⵏⴹ, ⴰⵙⴳⴳⵎⴹ ⴰⴱⵍⴰ ⴰⵙⵍⴽⵉⵏ.

TABLE DES MATIERES

Chapitre I : État d'art des Réseaux de capteurs sans fil (RCSFs)

1.1 Introduction ……..25

1.2 La ville intelligente……26

1.3 Définition d'un réseau sans fil……...26

1.4 Classification des réseaux sans fil…….27

1.4.1 Les WPAN (Wireless Personal Area Networks) ……...28

1.4.2 Les WLAN (Wireless Local Area Networks) …….....28

1.4.3 Les WMAN (Wireless Metropolitan Area Networks) …….....28

1.4.4 Les WWAN (Wireless Wide Area Networks)……....28

1.4.5 Les réseaux ad hoc……..29

1.5 Les Réseaux de Capteurs Sans Fil (RCSFs) ……29

1.5.1 Définition d'un capteur ……..29

1.5.2 Fonctionnement……...30

1.5.3 La communication entre les capteurs……..30

1.5.4 Architecture de communication……..31

1.5.5 Architecture d'un réseau capteur sans fil……....33

1.5.6 Anatomie d'un nœud capteur sans fil…….....34

1.6 Domaine d'application……...35

1.6.1 Les outils de simulation des RCSFs……....36

1.7 Les avantages et les contraintes d'un RCSF……...37

1.8 La sécurité des réseaux capteurs…….....38

1.8.1 Objectifs de la sécurité dans les RCSFs……...38

1.8.2 Issues majeures de la sécurité…….....39

1.8.3 Défis de la sécurité dans RCSFs…….....40

1.9 Conclusion……..43

Chapitre II : Les attaques sur les réseaux de capteurs sans fil

2.1 Introduction……44

2.2 Natures des attaques……44

2.2.1 Attaque passive……44

2.2.2 Attaque active……...44

2.3 Ses origines (interne /externe) ……45
2.3.1 Interne……...45
2.3.2 Externe……..45
2.4 Les différents types d'attaques…….46
2.4.1 Forced Delay…….46
2.4.2 Sybil……..46
2.4.3 Sinkhole……....47
2.4.4 Wormhole……..48
2.5 Les attaques connues et leurs solutions……...49
2.5.1 Jamming…….....49
2.5.2 Les différentes stratégies…….....49
2.5.2.1. Le Jamming constant…….....50
2.5.2.2. Le Jamming trompeur…….....50
2.5.2.3. Jamming aléatoire…….....50
2.5.2.4. Jamming réactif…….....50
2.5.3 Méthodes de défense…….....50
2.5.4 Hello Flood…….....50
2.5.4.1. Les solutions de défense.....51
2.6 L'attaque de retransmission sélective.....51
2.7 Attaque d'analyse de trafic.....52
2.7.1 Défense contre l'attaque d'analyse de trafic.....54
2.8 Conclusion…….....55

Chapitre III : Analyse des performances pour le protocole d'agrégation de données sécurisées dans les RCSFs

3.1 Introduction…….....57
3.2 Agrégation dans les RCSFs …….....57
3.3 L'agrégation et la sécurité des transmissions…….....59
3.3.1 Exigences de sécurité primaires…….....60
3.3.2 Types d'agrégation sécurisée de données.....60
3.3.3 Agrégation de données cryptées saut par saut.....61
3.3.4 Agrégation de données chiffrées de bout en bout.....62
3.3.5 Agrégation de données sécurisée non chiffrée.....64
3.4 Technique de sécurité…….....65
3.4.1 Cryptage d'homomorphisme…….....65

3.4.2 Signature numérique…….....65
3.4.3 MAC (code d'authentification de message) 65
3.4.4 Le tatouage numérique…….....65
3.4.5 Découpage de données…….....65
3.5 Analyse de la sécurité et des performances…….....66
3.5.1 Analyse de sécurité…….....66
3.5.2 Analyse des performances……67
3.6 Conclusions……..…….…….....68

Chapitre IV : Un schéma de signature sans certificat amélioré basé sur RSA pour les réseaux de capteurs sans fil

4.1 Introduction……..…….…….....69
4.2 Primitives cryptographiques utilisées dans les réseaux de capteurs.....70
4.3 Le chiffrement symétrique et asymétrique…….....70
4.3.1 Le chiffrement symétrique…….....70
4.3.2 Le chiffrement symétrique et asymétrique.....71
4.4 La signature numérique…..…….…….....72
4.4.1 Fonctions de Hachage…….…….....73
4.4.2 Fonctionnement de la signature numérique.....74
4.4.3 Certificats numériques…….…….....75
4.4.3.1. Autorité de Certification….....76
4.4.3.2. Vérification d'un certificat.....76
4.5 Le cryptosystème RSA…...…….…….....77
4.5.1 Génération des clefs…….…….....77
4.5.2 Les algorithmes de chiffrement, déchiffrement, signature et vérification de la signature : …...…….…….....79
4.5.3 Exemple d'illustration du cryptosystème RSA.....79
4.5.4 Schéma de signature numérique RSA.....80
4.5.4.1. Schéma de signature numérique RSA..... 80
4.6 Signatures sans certificat basé sur RSA.....81
4.6.1 Modèle de réseau.....81
4.6.2 Les étapes de signature sans certificat.....82
4.6.3 3 Proposition de schéma de signature sans certificat basé sur RSA.....82
4.6.4 Types des adversaires et leurs comportements de la signature sans certificat.83
4.6.4.1. Le modèle d'oracle aléatoire.....83

4.7 Analyse de sécurité.....84
4.7.1 Adversaires A_1 84
4.7.2 Adversaires A_2.....86
4.7.3 Intégrité.....88
4.7.4 Non-répudiation.....88
4.7.5 Les falsifications.....89
4.7.5.1. Les falsifications du message.....89
4.7.5.2. Falsification du message et de la signature.....89
4.7.6 Analyse et performance.....89
4.8 Conclusion…….....90

Chapitre V : Approche d'entités nommés de la langue amazighe pour les réseaux de capteurs sans fils

5.1 Introduction…….....91
5.2 Travaux connexes..…….....93
5.3 Traitement du langage naturel avec l'ontologie. 94
5.3.1 Portée temporelle.....95
5.3.2 Portée spatiale….....95
5.3.3 Portée caractéristique.....95
5.4 Traitement du langage naturel (TLN) 96
5.4.1 Arrêt de la suppression des mots.....96
5.5 Contexte linguistique…….....96
5.5.1 Brève description de la langue amazighe.....97
5.5.2 Tifinaghe Unicode…….....97
5.6 Morphologie de la langue amazighe…….....99
5.7 Défis et objectifs de la reconnaissance des nouvelles entités REN de l'amazighe….100
5.8 Corpus amazighe de test…….....101
5.9 Notre approche REN Amazighe…….....102
5.9.1 Système REN typique…….....102
5.9.2 Notre approche……..…….....103
5.9.3 Champs aléatoires conditionnels (CRFs) 105
5.9.4 Algorithmes utilisés…….....105
5.9.4.1. Algorithme pour l'identification des noms.....105
5.9.4.2. Algorithme d'identification REN …….....106

5.10 Ressources linguistiques pour REN amazighe…….....106
5.10.1 Création de répertoires géographiques…….…….....107
5.10.2 Mots déclencheurs……..…….…….....107
5.11 Résultats et discussion…….……..…….…….....107
5.12 Conclusion et perspectives……..…….....110

Chapitre IV : Méthode améliorée de cartographie probabiliste de Koblitz dans le cryptosystème à courbe elliptique : étude comparative et résultats

6.1 Introduction…….……..…….....112
6.2 Définition de la courbe elliptique……..…….....113
6.3 Définition de groupe sur des courbes elliptiques.. …….....116
6.4 La courbe elliptique sur les champs finis…….....116
6.4.1 Définitions et propriétés……..…….....116
6.4.2 Addition et doublement des points de la courbe elliptique.....117
6.4.2.1. Addition de deux points distincts…….....118
6.4.2.2. Doublement d'un point……..…….....118
6.4.3 Optimisation du doublement d'un point sur une courbe elliptique.....118
6.5 Le sous-groupe, l'ordre et le générateur d'un groupe de courbe elliptique.....119
6.5.1 Le sous-groupe……..…….....119
6.5.2 L'ordre d'un groupe de courbe elliptique.....119
6.5.3 Sélection d'un point générateur approprié pour générer un sous-groupe cyclique……..119
6.6 Cryptographie à courbe elliptique.. …….120
6.6.1 Problème de Logarithme discret de courbe elliptique (PLD).. ……121
6.6.2 Analyse de performance de RSA/DSA et ECC…….122
6.6.3 Comparaison entre les temps de calcul……...……..123
6.6.4 Cartographie (Mapping) des codes de texte en clair aux points d'une courbe elliptique…….……...……..…….123
6.6.4.1. La méthode de cartographie (Mapping) de Koblittz.. ……123
6.6.5 L'algorithme de cartographie de Koblitz modifié.. …..……125
6.7 Le chiffrement, déchiffrement et décodage utilisant la méthode de Koblitz modifié..126
6.7.1 Algorithme de chiffrement utilisant CE cryptographique……..……126
6.7.2 Algorithme de déchiffrement utilisant CE cryptographique ……..….....126
6.7.3 Décodage……....……...……...127

6.8 La langue amazighe et son système d'écriture……...……...127
6.9 Implémentation de chiffrement à l'aide des courbes elliptiques……...……...128
6.9.1 Mapping……...……..128
6.9.2 Chiffrement…….....……..130
6.9.3 Déchiffrement…….....……..130
6.9.4 Décodage…...…….....……..130
6.10 Travaux connexes…...…….....……..131
6.11 Résultats et comparaison…….....……..131
6.12 Conformité de la cartographie probabiliste de Koblitz.. …….....…….134
6.13 Analyse…….…….....……..…..134
6.13.1 Méthodologie…….....……..…..134
6.13.2 Résultats de mise en œuvre et comparaison…….....……..…...135
6.14 Conclusions…….....……..…....137

LISTE DES FIGURES

Figure 1.1 : La ville connectée s'organise autour de plusieurs axes……..……26
Figure 1.2 : les réseaux sans fil selon la zone de couverture……..27
Figure 1.3 : Architecture d'un réseau ad hoc……..29
Figure 1.4 : Capteurs en image……...30
Figure 1.5 : Fonctionnement du réseau de capteur sans fil……....30
Figure 1.6 : Exemple d'organisation des capteurs en "clusters…….....31
Figure 1.7 : Pile protocolaire……..32
Figure 1.8 : Collecte d'information 1…….....33
Figure 1.9 : collecte d'information 2……..34
Figure 1.10 : Structure d'un capteur [29]…….....35
Figure 1.11 : Application des RCSFs…….35
Figure 2.1 : Fonctionnement d'attaque Sybil……..46
Figure 2.2 : Schéma d'attaque Sinkhole……..…46
Figure 2.3 : Schéma d'attaque Sinkhole……47
Figure 2.4 : Schéma présente l'attaque Wormhole……..…48
Figure 2.5 : Principe d'attaque Jamming……49
Figure 2.6 : Attaque Hello Flood……51
Figure 2.7 : Attaque retransmission sélective……52
Figure 2.8 : Attaque analyse de trafic [43]……54
Figure 3.1 : Processus d'agrégation de données dans RCSF : *In-network processing*……..58
Figure 3.2 : Processus d'agrégation de données dans RCSF : *Elimination de la redondance*..58
Figure 3.3 : Processus d'agrégation de données par grappes……..59
Figure 3.4 : Arbre d'agrégation SDAP……61
Figure 3.5 : Scénario d'agrégation du protocole CDAP……62
Figure 3.6 : Model d'agrégation de données avec filigrane……63
Figure 3.7 : Architecture de découpage (taille de réseau u = 8, longueur de saut h_L = 1)……..66
Figure 4.1 : Schéma du Chiffrement symétrique……70
Figure 4.2 : Exemple de gestion des clés si N=4……72
Figure 4.3 : Chiffrement asymétrique : Aicha utilise la clé publique de Brahim afin de lui envoyer un texte chiffré. ……..71
Figure 4.4 : Une fonction de hachage H calcule un haché de n bits à partir d'un message arbitraire m……73
Figure 4.5 : La chance sur deux pour que deux personnes au moins soient nées le même jour de l'année ……...74
Figure 4.6 : Signature d'un message et sa vérification……75
Figure 4.7 : Anatomie d'une signature numérique avec certificat……..77

Figure 4.8 : Les étapes de la signature numérique de RSA……80
Figure 4.9 : Modèle de réseau du schéma proposé……82
Figure 5.1 : Exemple des annotations de TALN……..…...94
Figure 5.2 : Les étapes de traitement TLN pour extraire la portée temporelle, spatiale et caractéristique d'une requête en langage naturel……..…94
Figure 5.3 : Architecture typique d'un système REN……103
Figure 5.4 : Architecture de REN…….....104
Figure 5.5 : Performance Amazigh REN……...110
Figure 6.1 : Courbes elliptiques pour différentes valeurs des paramètres *a* et *b*. ……....113
Figure 6.2 : La loi d'addition sur les courbes elliptiques……...114
Figure 6.3 : L'addition de *P* à lui-même. ……..…115
Figure 6.4 : La droite (*L*) parallèle à l'axe des ordonnées traversant deux point *P* et *P'*……115
Figure 4.5 : Représentation graphique de la loi de groupe pour les courbes elliptiques……..116
Figure 6.6 : L'ensemble des points de la courbe elliptique E_{109} (–9,0): $y^2 = x^3 – 9x$ mod 109…117
Figure 6.7 : Le point générateur *G*(18,1), qui se répète cycliquement……..…120
Figure 6.8 : Estimer le niveau de sécurité des ECC et RSA en années MIPS……..….122
Figure 6.9 : Comparaison des performances de ECDSA-160, RS-1024 et DS-1024……123
Figure 6.10 : Interface émetteu (Cartographie/Mapping)……..….126
Figure 6.11: Interface du récepteur (décodage)……127
Figure 6.12. : Système d'écriture en langue amazighe Unicode en Afrique du Nord……127
Figure 6.13 : Insertion des paramètres de codage et de leurs résultats du message *M*……..129
Figure 6.14 : Insertion des paramètres de décodage du message *M* et de leurs résultats……..130
Figure 6.15 : Temps de cartographie de la méthode de Koblitz modifiées et la méthode de la [90].…….….132
Figure 6.16 : Temps de décodage de la méthode Koblitz modifiée et de la méthode [90]..…. 133
Figure 6.17 : Temps de codage et de décodage de la méthode Koblitz modifiée et [90]…..… 134
Figure 4.18 : Temps de codage et de décodage pour différentes valeurs : *a*, *b* et *p* de E*p* (*a*, *b*).134
Figure 6.19 : Temps de mappage et de chiffrement pour différentes longueurs de données de $E_{3946183951}$ (537680305, 1059676324)..……..136
Figure 6.20 : Temps de décryptage + décodage consommé par le schéma [91] et notre schéma137
Figure 6.21 : Consommation totale d'énergie pendant (cartographie + cryptages) et (décryptages + décodage) de notre approche et [91]……..….137

LISTE DES TABLEAUX

Tableau 1.1 : Les réseaux sans fil selon la portée…… ……28
Tableau 1.2 : Modèle TCP/IP (rappel)…….…….31
Tableau 3.1: Interaction entre la sécurité du réseau de capteurs sans fil et le processus d'agrégation de données……...60
Tableau 3.2 : Comparaison des méthodes d'agrégation de données……..…….64
Tableau 3.3 : Comparaison des protocoles d'agrégation de données……..……....67
Tableau 3.4 : Evaluations les performances des protocoles d'agrégation……..……67
Tableau 4.1 : Comparaison entre le chiffrement Symétrique et Asymétrique……...72
Tableau 4.2 : Complexité des meilleures attaques génériques……...74
Tableau 4.3 : Code de conversion……..79
Tableau 4.4 : Comparaison des propriétés de sécurité de notre schéma avec Zang *et al*…….....89
Tableau 5.1 : Système d'écriture choisi pour la translitération en latin…….98
Tableau 5.2 : Jeux des étiquettes grammaticales détaillées de la langue amazighe……101
Tableau 5.5 : Statistiques du corpus amazigh EN……108
Tableau 5.4 : Performances du système……...109
Tableau 6.1 : Cryptosystèmes à clé publique et leurs problèmes mathématiques. ……. 121
Tableau 6.2 : Estimation du niveau de sécurité des ECC et RSA/DSA en années MIPS……122
Tableau 6.3 : Comparaison des temps de calcul……...123
Tableau 6.4 : Conversion du message *M* en binaires. ……128
Tableau 6.5 : Conversion des blocs de messages en décimal……...128
Tableau 6.6 : Mapping des blocs m_i de messages *M* en points Pm_i (x_j, y_j). ……..…..129
Tableau 6.7 : Cartographie, chiffrement, déchiffrement et décodage pour le message *M* = ⵄⵣⵉ..131
Tableau 6.8 : Comparaison de cartographie, encodage / décodage entre la méthode Koblitz et [90] utilisant le message *M* = ⵄⵣⵉ.
Tableau 6.9 : Déchiffrement, décodage et énergie consommée pour différentes tailles de texte……..135

LISTE DES ALGORITHMES

Algorithme 4.1 : Génération des clés RSA ……77
Algorithme 4.2 : Chiffrement RSA……78
Algorithme 4.3 : Déchiffrement RSA……78
Algorithme 4.4 : Signature avec RSA ……78
Algorithme 4.5 : Vérification de la signature……78
Algorithme 5.1 : REN……104
Algorithme 5.2 : L'identification des noms……106
Algorithme 5.3 : L'identification REN……106
Algorithme 6.1 : Algorithme optimisé pour calculer un multiple d'un point sur une courbe elliptique (script python) ……118
Algorithme 6.2 : Mappage de Koblitz probabiliste d'un message en clair sur un point d'une courbe elliptique définie sur un champ fini……124
Algorithme 6.3 : Cartographie de Koblitz modifiée d'un message en texte clair à un point sur une courbe elliptique définie sur un champ fini……125
Algorithme 6.4 : Chiffrement utilisant ECC……126
Algorithme 6.5 : Déchiffrement utilisant ECC……126

LISTE DES ACRONYMES

AES	Advanced Encryption Standard
API	Alphabet Phonétique International
APTEEN	Adaptive de Threshold-sensitive Energy Efficient sensor Network protocol
BAN	Body Area Network
CDAP	Concealed Data Aggregation Privacy
CEISIC	Centre des Études Informatiques des Systèmes d'Information et de Communication
CH	Cluster Head
CLS	CertificateLess Signature
CoNLL	Conference on Computational Natural Language Learning
CRFs	Conditional Random Fields
DES	Data Encryption Standard
DoS	DoS Denial of Service
DPIS	Dummy packet injection Scheme
ECC	Elliptic Curve Cryptography
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
GPPS	Generic Privacy-Preservation Solutions
GPS	Global Positioning System
HCDA	Hilbert Curve Data Aggregation
HMAC	keyed-Hash Message Authentication Code)
HMM	Hidden Markov Model
HTTP	Hypertext Transfer Protoc
IE	Information Extraction
IoT	Internet of Things
IR	Information Retrieval
IRCAM	Institut Royal de la Culture AMazighe
ISO	International Organization for Standardization
KIPDA	k-Indistinguishable Privacy-preserving Data Aggregation
LPR	Location Privacy Routing Protocol
MAC	Message Authentification Code
MD5	Message Digest 5
MTC	Machine-Type-Communication
REN	Reconnaissance d'Entité Nommée
NIST	National Institute of Standards and Technologies
NLP	Natural Language Processing
NSDA	New Sensitive Data Aggregation
NTIC	Nouvelles Technologies de l'Information et de la Communication

OGC	Open Geospatial Consortium
OSI	Open Systems Interconnection
PA	Point Acces
PKG	Private-Key Generator
POS	Part Of Speech
QA	Question Answering
RAM	Random-access memory
RC5	Rivest's Cipher 5
RCSFs	Réseau de Capteur Sans Fil
RDF	Resource Description Framework
RFID	Identification par Radio-Fréquence
RSA	Chiffrement à clé publique de Rivest, Shamir et Adleman
SASPKC	Secure Aggregation using Stateful Public Key Cryptography Secure Data Aggregation Protocol
SDAP	Secure Data Aggregation Protocol
SDAW	Secure Data Aggregation Watermarking
SHA-1	Secure Hash Algorithm 1
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layers
SSN	Semantic Sensor Network
SVMS	Support Vector Machines
SWE	Sensor Web Enablement
TAL	Traitement Automatique des Langues
TCP	Transmission Control Protocol
TIC	Technologies de l'Information et de la Communication
TinyOS	Tiny Open System
TLN	Traitement du Langage Naturel
W3C	World Wide Web Consortium
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Networks
MAN	Wireless Metropolitan Area Networks
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network
WWAN	Wireless Wide Area Networks

INTRODUCTION GÉNÉRALE

1. Introduction :

Le nombre d'objets connectés a connu ces dernières années une croissance importante pour atteindre 15 milliards en 2018 pour l'Europe, avec une prévision entre 50 et 80 milliards pour 2020 [93]. Dans l'industrie, on parle de 4ème révolution industrielle ou d'internet des objets qui touche plusieurs domaines :

La santé: Récemment à cause de la pandémie du coronavirus Covid-19, qui est largement propagé dans la plupart des pays du monde, est signalé comme une pandémie contagieuse le 11 mars 2020 par l'OMS (Organisation mondiale de la santé).

Les recherches s'efforcent pour contrôler les malades porteurs du virus. Les chercheurs ont pensé d'utiliser les objets connectés comme les capteurs embarqués sur les smart phones. Cependant, une équipe de chercheurs marocains, composée d'ingénieurs et de médecins, ont annoncés le lancement d'une version initiale d'un « masque intelligent de détection automatique à distance » du Covid-19 [94]. Selon ces chercheurs, le masque, qui sert également de barrière de protection, contient une carte et des capteurs de température, d'humidité et de pression permettant de mesurer la pression et le cycle respiratoire, ainsi que le taux d'oxygène dans le sang (en le combinant avec un Oxymètre). Le coordinateur de l'équipe scientifique, a précisé que ce masque est relié via Bluetooth à l'application qui propose le « tracking » (traçage) des déplacements de l'utilisateur pour détecter son niveau de respect du confinement et de la distanciation sociale. Il a expliqué que cette application permet de remonter les données vitales du masque en les combinant avec les autres données d'auto-diagnostic, de détection intelligente par la voix et de suivi du comportement pour évaluer la probabilité d'infection, également de connaître les personnes fréquentées par l'utilisateur, tout en respectant la loi sur la protection des données personnelles.

Le transport : Avec des véhicules qui disposent en moyenne de 60 à 100 capteurs. Les voitures devenant de plus en plus « intelligentes », le nombre de capteurs devrait atteindre 200 capteurs par voiture, servant à l'optimisation du fonctionnement du véhicule ou au confort des passagers. Ces chiffres se traduisent par environ 22 milliards de capteurs utilisés dans l'industrie automobile par an d'ici 2020 [95].

Le militaire : Avec des vêtements intelligents équipés de capteurs et de circuits électroniques, l'internet des objets ne peut pas avoir lieu sans capteurs : qu'ils mesurent la température, la pression, la distance ou encore le mouvement, les informations collectées doivent être envoyées à une base de contrôle de données.

Les réseaux de capteurs sans fil ont élargi leur champ d'application, de collecter et de traiter des informations complexes issues de l'environnement (météorologie, étude des courants, de l'acidification des océans, de la dispersion de polluants, etc..)

L'objectif de ces nœuds capteurs est généralement de récolter des grandeurs physiques à partir de l'environnement où ils sont déployés (température, pression, humidité, luminosité, etc.), traiter l'information, et l'envoyer vers un (ou plusieurs) nœud(s) spécifique(s) dans le réseau appelé une entité de collecte ou Puits, ou alors vers une station de base.

Comme l'intérêt pour les réseaux de capteurs sans fil continue de croître, il en va de même pour la nécessité de mécanismes de sécurité efficaces. Étant donné que les réseaux de capteurs peuvent interagir avec des données sensibles et / ou fonctionner dans des environnements hostiles sans surveillance, il est impératif que ces problèmes de sécurité soient résolus dès le début de la conception du système. Donc la sécurité dans les réseaux de capteurs sans fil est devenue une préoccupation principale afin d'assurer une communication sécurisée entre les nœuds de capteurs statiques et mobiles.

Les nœuds des réseaux de capteurs sont généralement équipés d'une mémoire et une énergie limitée, l'énergie est considérée comme un véritable défi qui a motivé les chercheurs pendant la dernière décennie. Et d'ailleurs les travaux de recherche ont démontré que les solutions de sécurité proposées pour les réseaux sans fil (mobiles et adhoc), surtout celles basées sur l'utilisation de la cryptographie à clé publique ne peuvent pas être appliquées directement dans les réseaux de capteurs sans fil du fait des coûts de calcul importants engendrés.

Les RCSFs génèrent de gros volumes de données brutes, ce qui augmente la difficulté des applications à gérer et à interroger les données des capteurs. Les RCSFs sont normalement spécifiques à l'application, sans partage ni réutilisation des données de capteur entre les applications. Pour que les applications soient développées indépendamment de RCSF particuliers, les données des capteurs doivent être enrichies d'informations sémantiques. Les ontologies ou traitement du langage naturel (TLN) sont largement utilisées comme moyen de résoudre les problèmes d'hétérogénéité de l'information en raison de leur capacité à donner un sens explicite à l'information. Cependant les traitements de TLNs commencent à identifier les informations dans une source d'informations non structurée qui respecte la sémantique prédéfinie, comme les personnes, l'emplacement, etc. La reconnaissance des entités nommées (REN) à partir du texte en langage naturel lisible par ordinateur est une tâche importante pour l'extraction d'informations et peut être considérée comme une sous-discipline de l'intelligence artificielle.

Certaines langues ont éveillé beaucoup d'intérêt, notamment à travers les campagnes d'évaluation telles que CONLL (Tjong Kim Sang, 2002) pour l'espagnol et l'allemand, MUC (Grishman et Sundheim, 1996) pour l'anglais et le japonais, et ESTER (Galliano et al., 2009) pour le français. Toutefois, l'Amazighe étant une langue peu dotée en termes de ressources linguistiques informatisées, les travaux sur la reconnaissance des entités nommées sont une opportunité pour la valorisation de cette langue dans la société de l'information.

C'est dans cette optique que se situe notre travail et de développer un modèle de reconnaissance d'entité nommée pour la langue amazighe en utilisant une technique hybride, dont le but d'améliorer la précision du REN en langue amazighe qui servira à des applications plus spécifiques comme la gestion des données des RCSFs et de discuter des problèmes et les défis liés aux tâches de reconnaissance en NE pour la langue amazighe.

La langue amazighe fait partie des langues chamito-sémitiques ou encore appelé afro-asiatiques (Cohen 2007, Chaker 1989) qu'on soit au Maroc ou ailleurs. La langue Amazighe est maintenant une langue qui possède tous ses attributs : dotée d'une graphie officielle, un codage propre dans le standard Unicode, une grammaire, une orthographe et un vocabulaire très riche.

Globalement, la sécurité à été toujours la même, c'est d'assurer les services de sécurités tels que l'authentification, la confidentialité, l'intégrité, l'anonymat [96] et la disponibilité.

L'utilisation de courbes elliptiques en cryptographie est un sujet émergent mais important dans la recherche moderne sur la sécurité de l'information. Parce que les courbes elliptiques

offrent des avantages substantiels par rapport aux autres cryptosystèmes à clé publique en termes de taille de clé, de temps de traitement, de bande passante et de taille de code, elles sont rapidement acceptées pour leurs utilisations dans des applications légères comme les RCSF.

Assurer la sécurité dans les réseaux de capteurs permettra un gain de confiance des gens en cette nouvelle technologie ce qui va permettre d'élargir le domaine d'application et d'utilisation et ainsi faire des capteurs un outil qui peut nous accompagner dans notre vie quotidienne.

2. Contexte :

Les réseaux de capteurs sans fil deviennent de plus en plus populaires et leurs utilisations augmentent de jour en jour dans tous les domaines, la position de ces nœuds n'est pas obligatoirement prédéterminée, ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène observé.

Les réseaux de capteurs sans fil sont donc sujets à différents types de menaces et d'attaques telles que l'interception des données envoyées/reçues par le support sans fil et par la suite la possibilité de modifier et de rejouer les données. L'intrus peut également injecter, saturer ou endommager les équipements du réseau. Dans des applications critiques, de telles attaques peuvent être néfastes et peuvent engendrer des dégâts économiques et sécuritaires majeurs.

Dans ce contexte, la recherche continue à perfectionner le fonctionnement des futurs réseaux de capteurs. Pour améliorer sa compétence, les données échangées ne doivent pas être menacées ou modifiées par des événements extérieurs. En particulier, l'existence d'une politique de sécurité qui comporte la quadrature : la confidentialité, l'intégrité, la disponibilité des données et l'authentification [96] des différents nœuds de capteurs est essentielle. Les travaux de recherches se consacrent d'inventer ou de trouver des solutions qui éliminent les faiblesses de ces réseaux à partir des protocoles ou bien de produits de sécurité.

La mise en place des techniques capables d'éviter ce type de problème devient donc une nécessité dans le domaine de la recherche.

Une autre tâche basique utilisée par les RCSFs, est le traitement des données au niveau locale après sa capture et de les envoyer à un ou plusieurs points de collecte via une liaison sans fil, cette communication demande une sémantique attachée aux données échangées dans les RCSFs, pour faciliter la compréhension des données sensorielles.

Les réseaux de capteurs se caractérisent par deux caractéristiques principales [120]:

Premièrement, ils sont très denses de sorte que des centaines ou des milliers de nœuds peuvent être déployés dans des zones géographiques limitées. Ces nœuds renvoient une énorme quantité de données qui doivent être recherchées efficacement pour répondre aux requêtes des utilisateurs. Malheureusement, les techniques classiques de récupération d'informations ont montré de mauvaises performances dans la recherche de données de réseaux de capteurs car elles retournent de nombreux faux positifs / négatifs.

Deuxièmement, bon nombre des données saisies sont de nature analogue, ce qui rend la possibilité de trouver un terme spécifique assez bonne. La plupart des capteurs sont caractérisés par des mécanismes d'étalonnage similaires qui peuvent être décrits en utilisant des termes différents. Les techniques de recherche de correspondance de chaîne peuvent ne pas récupérer toutes les données pertinentes car différents mots / termes ont été utilisés qui ne correspondaient pas directement au terme. Par exemple, la mesure de la température est spécifiée avec le nom temp ou min temp et max temp ou lowtemp et high temp. L'humidité relative est un nom utilisé dans certains domaines et l'humidité dans d'autres domaines. Cela compromet les performances

du moteur de recherche. Une grande amélioration des performances des moteurs de recherche pourrait être obtenue si ces relations des domaines sont capturées et utilisées, et c'est exactement ce qu'une ontologie (Reconnaissance des Entités Nommées REN) peut faire.

La tâche de REN amazighes apparaît fondamentale pour diverses applications participant à l'analyse du contenu des textes amazighes, notre travail est de construire des extracteurs d'entités amazighes de très haute précision pour les catégories Personne, Lieu, Organisation et Divers qui minimiseraient les ambiguïtés, elles sont parmi les catégories les plus utilisées dans les systèmes de recherche d'informations à travers les domaines, elles peuvent donc être utilisées pour améliorer encore la précision du système REN et étendre sa portée grâce à l'apprentissage automatique.

La plupart des recherches se poursuivent pour sécuriser les RCSF avec des protocoles à clé symétrique, mais en même temps, la cryptographie à clé publique (PKC) a reçu très peu d'attention de la part des chercheurs.

Par rapport à RSA, ECC offre le même niveau de sécurité mais avec une taille de clé nettement plus petite, ECC est un bon candidat potentiel pour la sécurité des RCSF dans l'avenir proche [97]. Par exemple, une clé ECC-160 bits fournit le même niveau de la sécurité qu'une clé RSA-1024 bits et la clé ECC-224 bits offrent la même sécurité qu'une clé RSA-2048 bits. Des clés plus petites dans RCSF signifient un calcul plus rapide, une consommation d'énergie réduite et des économies de mémoire et de bande passante du nœud du capteur.

L'analyse détaillée et la modélisation mathématique de l'ECC ont été étudiées dans cette thèse.

3. Motivation :

Atteindre la sécurité dans un réseau de capteurs sans fil est une tâche difficile en raison du déploiement hostile et de la nature limitée des ressources des nœuds de capteurs. Les problèmes de sécurité liés à l'agrégation de données sont la confidentialité, la disponibilité, l'intégrité, l'authentification [96] et la mise à jour des données, etc. Ce sont tous les besoins de sécurité du réseau de capteurs sans fil. De nombreux protocoles d'agrégation sécurisés ont tenté de répondre à ces exigences ensemble. Quels sont les avantages et les inconvénients de chaque protocole ? et quels types de mécanismes d'authentification devraient être utilisés à l'avenir ?

Aujourd'hui les solutions de chiffrement à clés symétriques sont exploitables au sein des réseaux de capteurs et apportent des solutions réelles pour la sécurité du réseau. Cependant, si le chiffrement à clé symétrique est possible au sein des réseaux de capteurs, la sécurité totale de ce type de solution reste à démontrer. Car le chiffrement à clé symétrique ne garantit pas l'authentification des données comme la signature numérique des chiffrements à clés publiques, et RSA a déjà été implémenté dans diverses applications comme RCSF et le cloud computing, etc. utilisant les signatures agrégées sans certificat dans l'environnement de ressources informatiques limitées comme les RCSF, dans quelle mesure le modèle de sécurité, de schéma de signature sans certificat conserve les éléments essentiels de sécurité ? et quelles sont leurs types d'adversaires ?

La langue amazighe constitue un élément éminent de la culture marocaine et ce par sa richesse et son originalité. Elle à été négligée sinon écartée, pour des générations, en tant que source d'enrichissement culturel, la création de l'Institut Royal de la Culture Amazighe (IRCAM), suivie de la constitutionnalisation lui a permis désormais de jouir d'un statut meilleur [121]. Néanmoins, nous sommes tous conscients que l'officialisation n'est pas suffisante pour assurer la revitalisation de cette langue. D'ou le développement de la recherche dans les technologies de l'information et de la communication est devenue un support de vie social.

Une partie des ressources de contenu numérique amazighe est écrite sous forme de documents texte sous la forme de pages Web, de manuels de produits, d'articles de presse, de documents de recherche, etc. Le volume de ce contenu augmente à un rythme sans précédent, ce qui le rend très difficile à trouver pour les utilisateurs. La raison en est que ces ressources ne sont généralement pas correctement structurées en unités faciles à manipuler. Par conséquent, une analyse et une compréhension efficace du contenu textuel non structuré deviennent de plus en plus importantes. Cela a conduit à l'utilisation croissante des techniques de traitement du langage naturel (TLN) pour faciliter le traitement des documents, texte, non structurés. L'objectif fondamental de la recherche en TLN est de convertir un morceau de texte en une structure de données qui décrit sans ambiguïté et complètement la signification du texte en langage naturel.

Quelles sont les problèmes et les défis liés aux tâches de reconnaissance en NE pour la langue amazighe?, et quelle est sa particularité qui diffère considérablement des autres langues européennes ? ses ambigüités ? sa morphologie ?…

Avant que l'ECC ne devienne populaire, presque tous les algorithmes à clé publique étaient basés sur le cryptosystème RSA, l'algorithme de signature numérique (DSA) et l'échange de clés Diffie-Hellman (DH), des cryptosystèmes alternatifs basés sur l'arithmétique modulaire. RSA et ses amis sont toujours très importants aujourd'hui et sont souvent utilisés aux côtés de l'ECC. Cependant, la magie derrière RSA et ses amis peuvent être facilement expliqués par les implémentations approximatives qui peuvent être écrites assez facilement, alors, que les règles de base de la cryptographie utilisant ECC restent mystérieuses et incompréhensibles pour la plupart.

Comment peut-on utiliser une courbe elliptique cryptographique ? sur quelle partie faut-il intervenir pour rendre une EC cryptographique plus rapide avec l'énergie limitées ?

4. Contributions :

Cette thèse est basée sur six articles énumérés précédemment, également inclus dans leur intégralité à la fin de ce travail.

- Deux documents abordent différents aspects des problèmes de sécurité pour les réseaux de capteurs sans fil, et les attaques et les issues majeurs pour leur sécurité ;
- Le troisième document examine les performances des protocoles d'agrégation de données sécurisée dans les réseaux de capteurs sans fil.
- Le quatrième propose un schéma de signature sans certificat amélioré basé sur RSA pour les réseaux de capteurs sans fil, et que ce schéma est sécurisé dans le modèle d'oracles aléatoires et que sa sécurité est étroitement liée au problème de logarithme discret.
- Le cinquième document présente une approche d'entités nommés de la langue amazighe et son intérêt pour les RCSFs au cours de leurs échanges d'informions.
- Le sixième document présente une étude comparative utilisant une méthode améliorée de cartographie probabiliste de Koblitz dans le cryptosystème à courbe elliptique.

Au niveau du chapitre II : Après avoir présenter les origines et les différents types d'attaques qui peuvent perturber ou détruire un réseau de capteur sans fil, nous avons rappelé quelques types d'attaques connues qui visent un des objectifs de la sécurité comme l'authentification c'est le cas pour l'attaque Hello Flood. Dans ce sens nous avons cité les méthodes et les mécanismes utilisés pour se défendre contre ces attaques. Nous avons décrit

l'attaque d'analyse de trafic et les défenses possibles contre cette attaque, mettant en place de nouvelles routes et générant du faux trafic.

- Découvrir les origines et les différents types d'attaques qui peuvent perturber ou détruire un réseau de capteur sans fil, comme : Forced Delay, Sybil, Sinkhole et Wormhole.
- Présentation les principaux types d'attaques connus, leurs fonctionnements et leurs solutions de défense : Jamming, Hello Flood et Attaque d'analyse de trafic.

Au niveau du chapitre III : Nous avons examinés les protocoles d'agrégation de données sécurisée, à la pointe de la technologie. Ces protocoles peuvent être classés et analysés lors de l'agrégation de données chiffrées de saut par saut, bout en bout, et l'agrégation sécurisée de données non chiffrées, sur la base du mécanisme de mise en œuvre de l'agrégation sécurisée de données. Dans le but d'évaluer l'efficacité des protocoles mentionnés. Enfin, nous avons discuté de certaines questions à étudier à l'avenir.

- Comparaison et évaluation l'efficacité des protocoles d'agrégation de données sécurisée.
- Analyse de la sécurité et des performances, pour différents protocoles d'agrégation de données sécurisées basés sur les technologies de sécurité et les fonctions d'agrégation.

Au niveau du chapitre IV: Présente une analyse de sécurité d'un schéma de signature sans certificat amélioré basé sur RSA (RSA-CLS) pour les réseaux de capteurs sans fil. La complexité de la sécurité de notre contribution est étroitement liée aux problèmes de l'hypothèse forte de RSA (Strong RSA Assumption) et logarithme discret (DLP : Discrete Logarithm Problem) [68].

- Notre schéma RSA-CLS utilise sept algorithmes temporels polynomiaux pour la génération de signatures. Le niveau de sécurité de notre schéma à été sûr sous certaines conditions bien étudiées contre les adversaires de type A_1 et A_2, avec une probabilité non négligeable.
- Notre schéma a de meilleures performances que les travaux présentés par Zang et al [73] dans un réseau de capteurs sans fil et fournissent presque tous les objectifs de sécurité essentiels.

Au niveau du chapitre V : Ce travail le besoin de reconnaissance d'entité nommée NE pour la langue amazighe et discute des problèmes et des défis liés aux tâches de reconnaissance en NE pour la langue amazighe. Explore également diverses méthodes et techniques qui sont utiles pour la création de ressources d'apprentissage et de lexiques qui sont importants pour l'extraction d'EN à partir d'un langage naturel non structuré.

L'objectif principal de notre travail est de construire des extracteurs d'entités amazighes de très haute précision pour les catégories Personne, Lieu, Organisation et Divers qui minimiseraient la sortie bruyante (entités et leurs relations) afin de les appliquer sur RCSFs dans les travaux prochains.

- Construire des extracteurs d'entités amazighes de très haute précision pour les catégories Personne (93%), Lieu (97%), Organisation (74%) et Divers (65%) qui minimiseraient la sortie bruyante (entités et leurs relations) suivant les algorithmes CRF (Conditional Random Fields).
- Tentative de développer un modèle de reconnaissance d'entité nommée pour la langue amazighe en utilisant une technique hybride. Le but de ce modèle est d'améliorer la précision du REN en langue amazighe introduit par différentes approches dans la littérature.

Au niveau du chapitre VI : Ce chapitre fournit les connaissances et les conventions de base pour comprendre ce qu'est la cryptographie à courbe elliptique.

Notre contribution présente une amélioration de la méthode de cartographie probabiliste de Koblitz en utilisant un exemple de message écrit en caractères Tifinaghe Unicode.

- Proposition d'un nouvel algorithme d'optimisation pour la multiplication scalaire sur les courbes elliptiques. Ce qui réduit le nombre d'opérations ponctuelles et améliore remarquablement l'efficacité de calcul de la multiplication scalaire. Moins d'opérations ponctuelles garantissent un temps de calcul de clé plus rapide.
- Développement d'un nouvel algorithme de codage (mapping) de Koblitz pour faciliter la recherche des racines carrées de l'équation EC correspondant aux blocs du message M dans un temps réduit.
- Présentation de la définition de la langue amazighe et son système de codage et d'écriture au Maroc, et leur implémentions dans le processus de chiffrement et déchiffrement à l'aide de ECC.
- Pour évaluer l'efficacité de notre méthode, nous l'avons comparée à deux méthodes. L'une basée sur un point générateur *G* d'un sous-groupe de EC avec une matrice qui change la position des points [90]. et la seconde utilise un schéma de code ECC hybride codé par l'ADN [91]. Les résultats ont montré que les calculs de temps de chiffrement et de déchiffrement du texte et l'énergie consommée durant ces deux processus sont réduits.

5. Organisation de la thèse :

Ce manuscrit est composé de six chapitres énumérées précédemment. Les documents abordent différents aspects des problèmes énoncés dans la section « motivation ».

Nous nous intéressons à trois problèmes : le premier sur la recherche dans le domaine de la sécurité des RCSF, nous cherchons à définir des mécanismes et des solutions qui répondent à la politique de sécurité qui comporte la confidentialité, l'intégrité, la disponibilité des données et l'authentification des différents nœuds de capteurs.

Le deuxième problème concerne l'extraction d'entités nommées (EN) qui est une étape importante pour le traitement de contenu non structuré. Les données non structurées sont opaques sur le plan des calculs. Les ordinateurs (machines) ont besoin de données transparentes pour le traitement. L'extraction d'informations (EI) ajoute un sens aux données brutes afin qu'elles puissent être facilement traitées par les ordinateurs.

Le troisième problème consiste en l'étude de la sécurité basée sur les courbes elliptiques.

Dans le premier chapitre une description générale de réseaux de capteurs sans-fil, leurs domaines d'applications, leur architecture et leur contrainte sont étudiée.

Nous donnons par la suite quelques notions importantes sur le fonctionnement et les composantes d'un capteur sans fil utilisées ainsi que les principaux avantages et inconvénients offerts en expliquent les contraintes influençant l'architecture ainsi que des divers domaines qui les utilisent.

Par la suite, nous introduisons dans le chapitre 2, l'importance de la sécurité dans les RCSF, les attaques et les issues majeurs pour leur sécurité, nous citons les solutions de défense capables de détecter quelques dangereuses attaques contre les RCSFs en se concentrant sur leurs principes de fonctionnement et des discutions de certaines questions à étudier à l'avenir.

Le chapitre 3 donne une analyse des performances pour le protocole d'agrégation de données sécurisées dans les réseaux de capteurs sans fil : les types d'agrégation sécurisée de données, les techniques de sécurité afin d'analyser leurs performances.

Le chapitre 4 décrit les primitives cryptographiques utilisées dans les réseaux de capteurs sans fils, qui couvre la signature numérique et la fonction de hachage. Ensuite on présente le travail connexe de notre contribution basé sur RSA, suivi d'une analyse de sécurité dans des oracles aléatoires, avec une analyse et performance.

Dans le chapitre 5 on commence par l'utilisation des entités nommées pour les RCSFs, ensuite on explique l'intérêt de reconnaissance en EN pour la langue amazighe et on discute les problèmes et les défis liés aux tâches de reconnaissance en EN pour la langue amazighe. Dans cette partie on explore également diverses méthodes et techniques qui sont utiles pour la création de ressources d'apprentissage et de lexiques qui sont importants pour l'extraction de EN à partir de texte non structuré en langage naturel.

Le chapitre 6 aborde une méthode améliorée de cartographie probabiliste de Koblitz dans le cryptosystème à courbe elliptique et commence de fournir une définition claire des courbes elliptiques et son arithmétique sur les nombres réels et sur les champs finis, avec un nouvel algorithme optimisé de multiplication scalaire d'CE. Par la suite on décrit le système cryptographique basé sur une courbe elliptique utilise également l'approche proposée de codage et de décodage et son processus de chiffrement/déchiffrement. Une section a été réservée pour la langue amazighe et son système de codage et d'écriture au Maroc, afin de mettre une mise en œuvre de la méthode proposée (Koblitz).

Enfin, nous terminerons par une conclusion générale, et nous énoncerons des perspectives de recherche.

Chapitre

1

État d'art des Réseaux de Capteurs Sans Fil (RCSFs)

1.1 Introduction:

De nombreuses avancées techniques et technologiques donnent des nouvelles perspectives dans le domaine des télécommunications comme un nouveau type de composants « les capteurs électroniques » qui sont constitués dans la majeure partie des cas d'un microprocesseur, d'une mémoire vive, d'une interface radio et d'une source d'énergie. L'assemblement de ces capteurs est appelé " réseaux de capteurs sans fil " (RCSFs).

On peut définir un réseau de capteurs sans fil (RCSF) comme un ensemble de nœuds dédiés à la collecte d'information, en même temps, capables de communiquer entre eux afin de réaliser différentes tâches. La facilité de déploiement de ce type de réseau constitue un atout qui les rend facilement intégrables dans une grande variété d'applications : militaires, environnementales, domestique, industrielles, etc. Étant de petite taille, ces nœuds, ont la possibilité d'être déployer dans des endroits à accès difficile ou dangereux pour l'être humain. Cependant ce type de réseau est vulnérable et nécessite une très grande sécurisation surtout au niveau des nœuds, cela consiste à l'utilisation de nouveaux protocoles notamment la gestion des clés qui est une des majeures solutions pour augmenter la sécurité des réseaux de capteur sans fil.

La problématique principale des réseaux est sans aucun doute le manque de sécurité résultant de la difficulté d'appliquer les protocoles sécuritaires ordinaires sur les réseaux de capteurs, car ces derniers sont limités par leur batterie et leur puissance de calcul. La sécurité est indispensable dans ce cas vu que ces réseaux sont déployés dans des zones hostiles, ce qui les rendent des proies faciles pour les attaquants qui peuvent provoquer différents types d'attaques. Assurer la sécurité dans les réseaux de capteurs permettra le gain de la confiance des utilisateurs dans cette nouvelle technologie ce qui va permettre l'élargissement du domaine d'application et d'utilisation et ainsi faire des capteurs un outil qui peut nous accompagner dans notre vie quotidienne.

Notre projet entre dans le cadre de l'étude des réseaux de capteurs sans fil précisément la sécurité des RCSFs ainsi que la cryptographie symétrique. Comme nous allons le voir dans la suite de ce manuscrit, le domaine des RCSFs est très compliqué au niveau de la sécurité à cause essentiellement des attaques rencontrées par ce type de réseau et l'absence d'infrastructures fixes et de toute administration centralisée vu que la communication est ouverte et basée sur des nœuds.

Ce chapitre est consacré à une présentation générale sur la notion de capteurs et les réseaux de capteurs sans fil et des notions importantes sur leur fonctionnement et leurs composantes utilisées ainsi que les principaux avantages et inconvénients offerts en expliquent les contraintes influençant l'architecture ainsi que des divers domaines qui les utilisent.

1.2 La ville intelligente :

L'internet des objets connectés (Internet of Things (IoT)) peut être défini comme le concept qui permet de connecter dans un même réseau global des appareils ou objets divers (capteurs, ordinateurs, appareils électroniques...), identifiables de manière unique, grâce à des communications filaires et sans fil.

D'après [1], plus de 70% de la population vit aujourd'hui dans des zones urbaines en Amérique du Nord et du Sud, en Europe et en Océanie. Les projections actuelles prévoient que la population urbaine sera majoritaire sur l'ensemble des continents en 2035 et atteindra 68% en valeur moyenne à l'horizon 2050.

L'augmentation de la population urbaine et la densification des villes introduisent de nouveaux défis de gestion et de gouvernance des espaces urbains : la gestion du trafic et de la signalisation, des parcs de stationnement, des déchets, de la relève des compteurs d'eau, d'électricité, de gaz, etc [2]. Les méthodes de gérance traditionnelle, basées sur l'intervention manuelle d'agents, deviennent inadéquates en regard des densités de population urbaines observées, l'agglomération de Tokyo représente par exemple 42.8 millions d'habitants en 2018 [3].

Les réseaux de capteurs apportent une solution technique à l'automatisation de ces tâches de gérance, en permettant par exemple de télé-relever les compteurs ou de renseigner sur la disponibilité d'une place de parking.

Pour la création et la meilleure gestion de la ville intelligente, les RCSFs sont utilisés comme une technologie spécifique avec un but de créer un réseau réparti de noyaux de capteurs et actionneurs intelligents qui peuvent mesurer plusieurs paramètres intéressants. Toutes les données sont captées et transmises en temps réel aux utilisateurs ou aux autorités concernés.

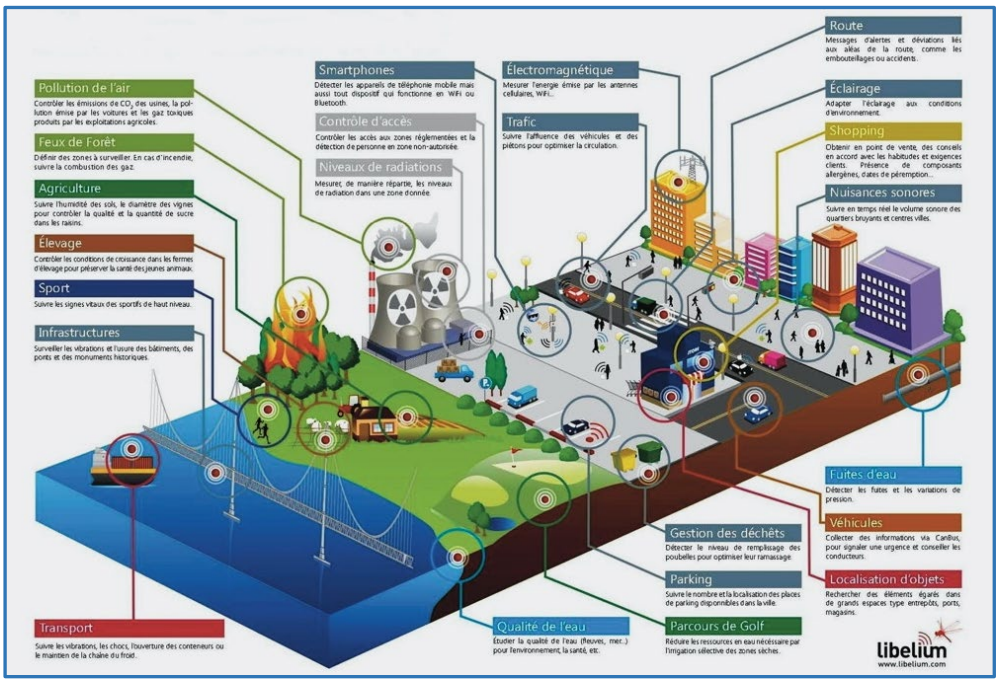

Figure 1.1 : La ville connectée s'organise autour de plusieurs axes

1.3 Définition d'un réseau sans fil

Un réseau sans fil est un réseau informatique qui aide à circuler les informations dans un système de communication sans contrainte de câblage [4]. Ce type de réseaux donneront la

possibilité de rester connecter même s'il a un périmètre géographique plus étendu « mobilité » entre les bureaux, les salles de réunions et les laboratoires. Ils donneront aussi la possibilité de répondre à la difficulté des grands sites où le câblage est trop coûteux ; campus, usines, etc [5].

La transmission des informations dans les réseaux sans fil est faite soit par liaison infrarouge soit onde radio, mais la méthode la plus intéressante c'est l'onde radio grâce à sa plus large couverture géographique et son débit très élevé. Pour trouver la technologie de transmission la plus adapté, d'une part à l'aide de la fréquence d'émission utilisée et d'autre part par le débit et la portée [5].

La construction des réseaux sans fil a pour but de répondre à deux besoins principaux :

- Assurer des transmissions dans des cas ou la pose de câbles est impossible.
- Permet de transmettre des données dans les applications mobiles.

En outre, la mise en place et l'installation des réseaux sans fil sont simples, ce qui simplifie leur développement. Nous rappelons par ailleurs, la difficulté de la réglementation qui concerne les différents domaines d'application (militaires, scientifiques, amateurs...) qui nécessite des accords après avoir donné les puissances maximales d'émission et les règles de coexistence dans les bandes de fréquences.

1.4 Classification des réseaux sans fil

1.4.1 Les WPAN (Wireless Personal Area Networks)

Dans cette catégorie de réseau personnel sans fil (appelé également réseau individuel sans fil ou réseau domestique sans fil) on retrouve les réseaux sans fil à l'échelle humaine dont la portée maximale est limitée à quelques dizaines de mètres autour de l'usager (bureaux, salles de conférence...). On y trouve les standards tels que le Bluetooth [6] connu aussi sous le nom de la norme IEEE802.15.1, son débit théorique est supérieur à 24 Mbits/s pour une portée de 100 mètres [7]. ZIGBEE [8] permet la communication machine-machine, avec une très faible consommation électrique et des coûts très bas [9], ce qui le rend particulièrement facile à l'intégrer dans des petits appareils. Enfin, HomeRF [10] qui a un débit d'environ 10 Mbits/s avec une portée avoisinant les 100 mètres.

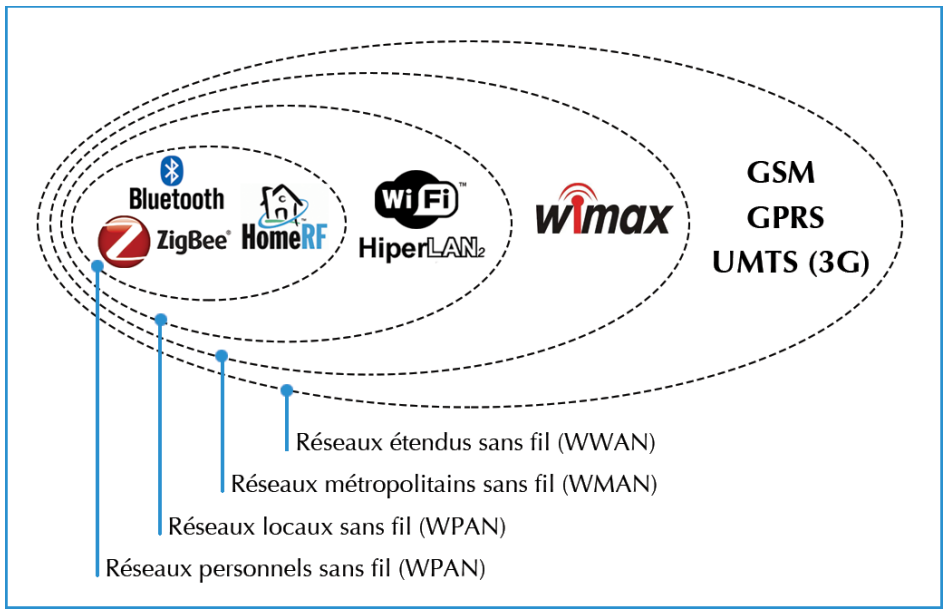

Figure 1.2 : les réseaux sans fil selon la zone de couverture

1.4.2 Les WLAN (Wireless Local Area Networks)

C'est la catégorie des réseaux locaux sans fil dont la portée va jusqu'à 500 m, pour les applications couvrant un campus, un bâtiment, un aéroport, un hôpital, etc. On y trouve les standards tels que IEEE802.11[11] (Wi-Fi pour Wireless Fidelity) permet de relier des ordinateurs portables, des assistants personnels (PDA : Personal Digital Assistant.), des objets communicants ou même des périphériques à une liaison haut débit (de 11 Mbit/s théoriques ou 6 Mbit/s réels en 802.11b à à 54 Mbit/s théoriques jusqu'a 1 Gb dans la norme 802.11ac [12]) sur un rayon de plusieurs dizaines de mètres, HIPERLAN [13] qui offre un débit de 20 Mbits/s, mais la version Hiperlan2 permet d'atteindre 54 Mbits/s sur une bande de fréquence de 5GHz.

1.4.3 Les WMAN (Wireless Metropolitan Area Networks)

Plus connus sous le nom de Boucle Locale Radio (BLR) [14], ce type de réseau utilise le même matériel que celui qui est nécessaire pour constituer un WLAN mais peut couvrir une plus grande zone de la taille d'une ville tout en économisant le cout élevé de la pose du câblage fibre ou cuivre. Les WMAN sont basés sur la norme IEEE 802.16 [15]. La norme 802.16 est généralement appelée Wimax, Il exploite une bande de fréquence de 2 à 11Ghz. La portée peut atteindre 122 km dans la normal 802.11e [16], offrant un débit jusqu'à 1 Gbit/s stationnaire et 100 Mbit/s en mobile grande vitesse dans la norme 802.16m [17].

1.4.4 Les WWAN (Wireless Wide Area Networks)

Le réseau étendu sans fil est la catégorie de réseaux cellulaires mobiles dont la zone de couverture est très large, à l'échelle mondiale. WWAN sont des réseaux basés sur des infrastructures comme MSC et les stations de base BTS pour permettre à l'utilisateur mobile des connexions sans fil sur le réseau public ou privé distants [14]. Dans cette catégorie, on peut citer le GSM (Global System for Mobile Communications) [18] et ses évolutions GPRS [19] (General Packet Radio Service), EDGE (Enhanced Data GSM Environment), l'UMTS [20] (Universal Mobile Telecommunication System), ainsi que la technologie LTE (Long Term Evolution) et LTE-Advanced. Le débit dans cette dérnière peut atteindre ou dépasser 1 Gbit/s [21].

Catégorie	Portée max	Débit	Normes	Usages
WPAN	Jusqu'à 100 m	Environ10 Mbits/s	IEEE 802.15 (Bluetooth) NFC,	Réseau particulier (petits appareil, la communication machine-machine..)
WLAN	500 m	Allant jusqu'à 450 Mbit/s	IEEE 802.11n	Réseaux internes, propres à un bâtiment (soit comme réseau d'entreprise ou domestique): Bâtiment, Aéroport, Hôpital..
WMAN	50 à 122 Km	de 100 Mbit/s a 1Gbit/s	IEEE 802.16 (Wimax) IEEE 802.11e	Régional, National Interconnecte plusieurs villes

WWAN	Plusieurs centaine de Kms	1 Gbit/s	Basé sur des technologies cellulaires	Régional, National Interconnecte plusieurs villes

Tableau 1.1 : Les réseaux sans fil selon la portée

1.4.5 Les réseaux ad hoc :

Les réseaux ad hoc sont l'un des réseaux sans fil (ondes radio) qui ne nécessitent pas d'infrastructure prédéfinie, ils ont deux types d'architecture soit cellulaire soit WIFI et grâce à ces derniers ils peuvent offrir une grande puissance et une bonne stabilité. L'une des principales caractéristiques de ce type des réseaux est la mobilité des nœuds. Sur la figure 1.3 qui montre la topologie des réseaux ad hoc, on remarque que chaque nœud est capable de communiquer directement avec ses voisins par lesquels ils passent pour communiquer avec des nœuds plus éloignés ce qui nécessite l'emploi d'un routage interne par des nœuds intermédiaires afin de faire acheminer les paquets de messages à la bonne destination [22].

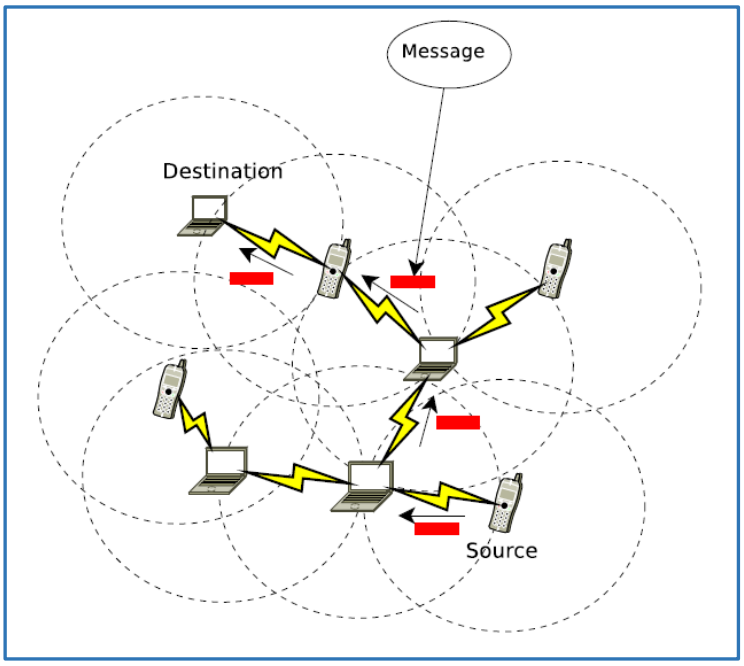

Figure 1.3 : Architecture d'un réseau ad hoc

1.5 Les Réseaux de Capteurs Sans Fil (RCSFs)

Les réseaux de capteurs sans fil (*RCSFs*) qui change radicalement la façon de concevoir les systèmes de surveillance de très grande échelle. En effet, les avancées récentes dans la micro-électronique et la communication sans fil ont permis le développement des capteurs de plus en plus petits à faible coût fonctionnant sur batterie et caractérisés par leurs ressources limitées en termes de bande passante, débit, énergie, zone de couverture, capacités de traitement et de stockage. Actuellement, les nœuds *RCSF* ont tendance à intégrer le monde d'internet pour profiter de ses atouts dans le cadre des technologies, *IdO* et *Web des Objets* « *WoT* '*Web of Things*' » [23] [24] [25].

1.5.1 Définition d'un capteur :

C'est un dispositif qui transforme une grandeur physique observée (température, pression, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur) [26]. Pour

cela, il possède au moins un transducteur dont le rôle est de convertir ce type de grandeurs.

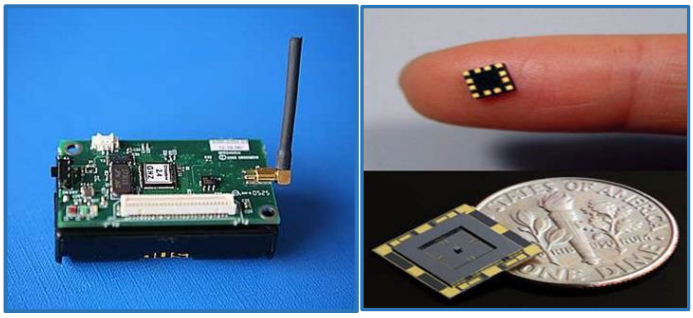

Figure 1.4 : Capteurs en image

1.5.2 Fonctionnement

Le fonctionnement principal du réseau de capteur sans fil est classique, dans une zone géographique appelée : zone de captage, plusieurs nœuds dispersés aléatoirement et ~~ce~~ communiquent entre eux via des liens radio pour le partage d'information et le traitement coopératif jusqu'à la surveillance d'un phénomène et la récolte des données de manière autonome [27]. A l'aide d'un routage multi saut les données captées peuvent être acheminé vers un nœud considéré comme un point de collecte, appelée nœud-puits (ou sink), ce nœud va lui-même transférer les données collectées via internet ou satellite à un ordinateur central gestionnaire de taches pour leurs traitements. Offrant à l'usager la permission d'adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits [28]. La figure 1.5 représente le fonctionnement de ce type de réseau.

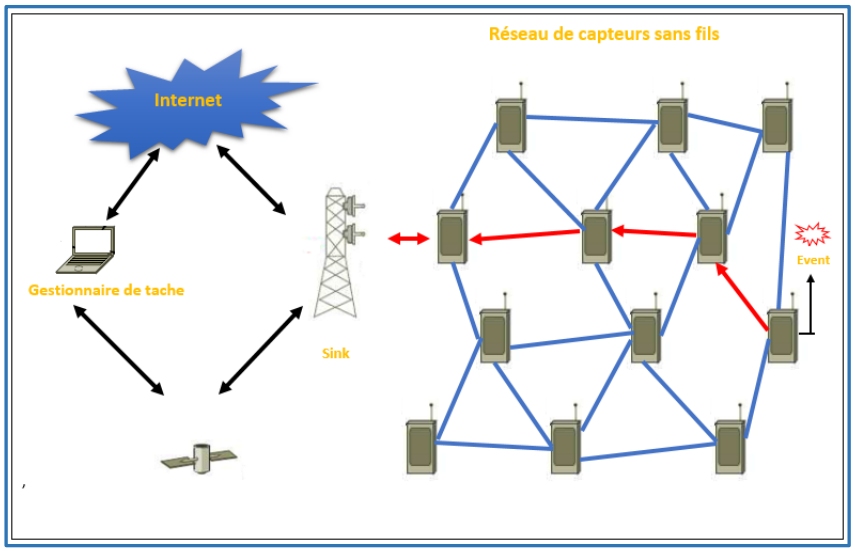

Figure 1. 5 : Fonctionnement du réseau de capteur sans fil

1.5.3 La communication entre les capteurs

Au niveau de la communication entre les capteurs, on a besoin de régler plusieurs paramètres comme les applications et l'objectif du réseau pour obtenir la meilleure stratégie, cette communication peut être prise par plusieurs façons [29] :

- **La démarche évènementielle :** dans ce type de démarche il y a une consommation d'énergie car le capteur et en mode repos puisque l'envoie des données se fait suite à une requête ou capture de donnée.

- **La centralisation des données :** ce type donnera une organisation des capteurs dans des ensembles clusters puisqu'il permet l'interception des évènements, et les envoyer au clusters-Head, qui reçoit toutes les informations transmises par les autres capteurs clusters (voir figure 1.6).

- **La distribution des données :** dans ce type il y a une organisation en mailles des capteurs car il permet de localiser les capteurs voisins, effectuer des calculs, prendre des décisions collectivement.

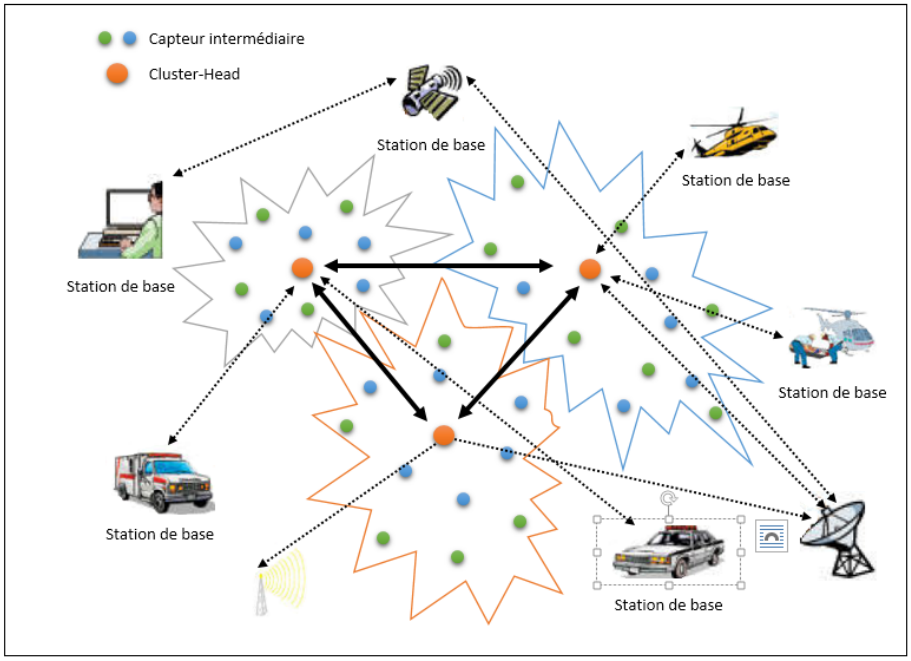

Figure 1.6 : exemple d'organisation des capteurs en "clusters

1.5.4 Architecture de communication

Cette propriété est la plus importante dans les RCSFs c'est pour cela elle exploite une pile protocolaire similaire (regroupe dans un modèle en couches) de celle de modèle OSI (Open Systems Interconnection) ainsi qu'elle vise les différents plans de gestion en utilisant les distincts standards protocoles de communication sans fil, reprend donc le modèle TCP/IP. [30]

Exemple :

5	Application	HTTP, FTP, SSH
4	Transport	TCP (Transmission Control Protocol), UDP
3	Réseau	IP
2	Liaison	IEEE 802.11 (CSMA/CA)
1	Physique	Ondes électromagnétiques

Tableau 1.2 : Modèle TCP/IP (rappel)

- **Modèles en couches :**

L'approche modèle vient pour standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que 3 couches pour la gestion de la puissance d'énergie, la gestion de la mobilité ainsi

que la gestion des tâches (interrogation du réseau de capteurs). Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Elle utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.

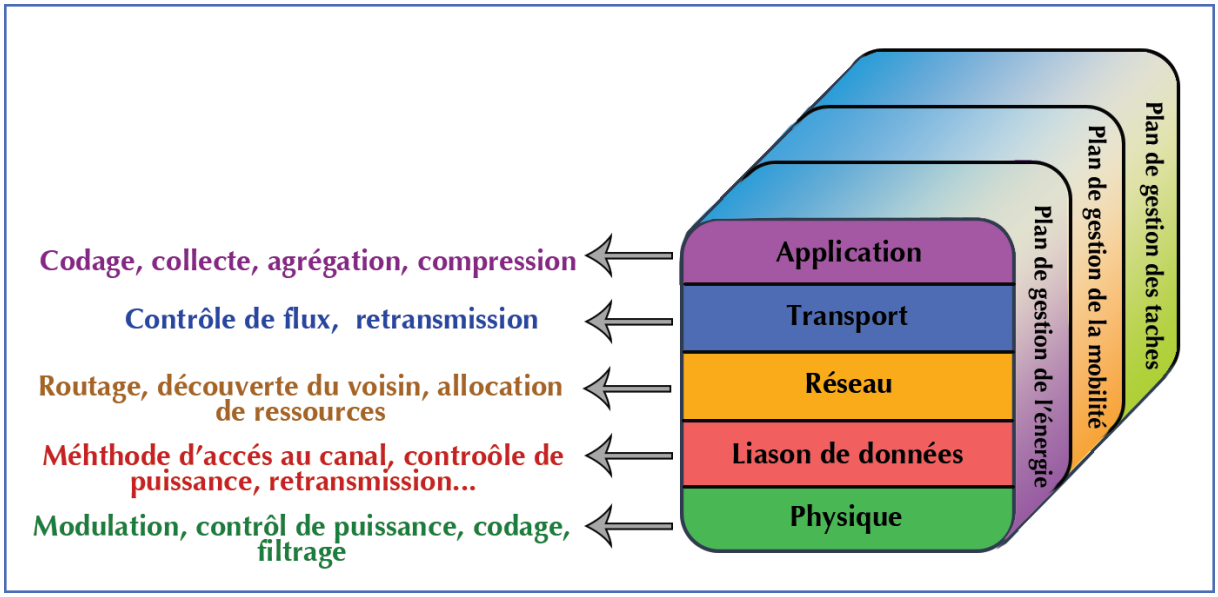

Figure 1.7 : pile protocolaire

- **Les couches existent :**

- **La couche physique :** Spécifications des caractéristiques matérielles, des fréquences porteuses, etc...

- **La couche liaison :** Spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au media, Elle assure la liaison point à point et multi-point dans un réseau de communication.

- **La couche réseau :** Dans la couche réseau le but principal est de trouver une route et une transmission fiable des données, captées, des nœuds capteurs vers le puits « sink » en optimisant l'utilisation de l'énergie des capteurs. Ce routage diffère de celui des réseaux de transmission ad hoc sans fils par les caractéristiques suivantes : Il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds. Les applications des réseaux de capteurs exigent l'écoulement des données mesurées de sources multiples à un puits particulier. Les multiples capteurs peuvent produire de mêmes données à proximité d'un phénomène (redondance). Les nœuds capteurs exigent ainsi une gestion soigneuse des ressources. En raison de ces différences, plusieurs nouveaux algorithmes ont été proposés pour le problème de routage dans les réseaux de capteurs. [30].

- **La couche transport :** Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

- **La couche application :** Elle assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

- **Plans de gestion :** Les plans de gestion d'énergie, de mobilité et de tâche contrôlent l'énergie, le mouvement et la distribution de tâche au sein d'un nœud capteur. Ces plans aident

les nœuds capteurs à coordonner la tâche de captage et minimiser la consommation d'énergie. Ils sont donc nécessaires pour que les nœuds capteurs puissent collaborer ensemble, acheminer les données dans un réseau mobile et partager les ressources entre eux en utilisant efficacement l'énergie disponible. Ainsi, le réseau peut prolonger sa durée de vie.

- **Plan de gestion d'énergie :** Contrôle l'utilisation de la batterie. Par exemple, après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage. L'énergie restante est réservée au captage.

- **Plan de gestion de mobilité :** détecte et enregistre le mouvement du nœud capteur. Ainsi, un retour arrière vers l'utilisateur est toujours maintenu et le nœud peut garder trace de ses nœuds voisins. En déterminant leurs voisins, les nœuds capteurs peuvent équilibrer l'utilisation de leur énergie et la réalisation de tâche.

- **Plan de gestion de tâche :** balance et ordonnance les différentes tâches de captage de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de captage au même temps ; certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie. [30]

1.5.5 Architecture d'un réseau capteur sans fil

Les réseaux de capteurs sans fil sont des réseaux spontanés constitues de nœuds déployés en grand nombre en vue de collecter et de transmettre des données vers un ou plusieurs points de collecte, et ce de façon autonome.

Les nœuds capteurs composants le réseau possèdent généralement de faibles capacités de calcul, de mémoire et d'énergie, l'accès au medium radio étant l'élément le plus couteux.

Il y a deux façons de collecter les informations : à la demande ou suite à un événement.

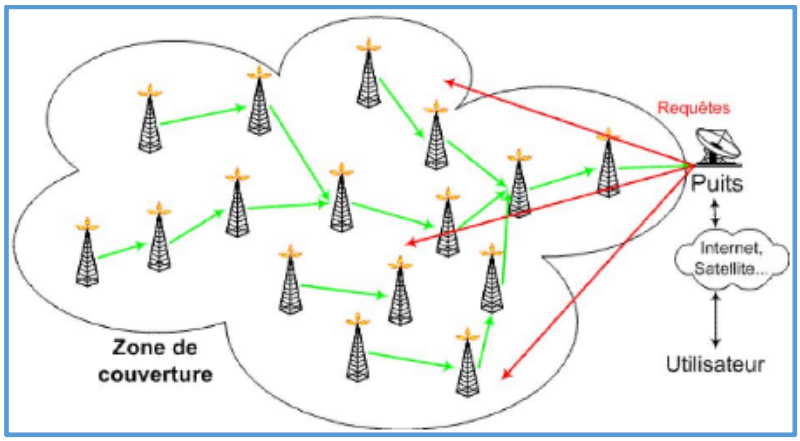

Figure 1.8 : collecte d'information 1

Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment *t*, le puit émet des brodcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts.

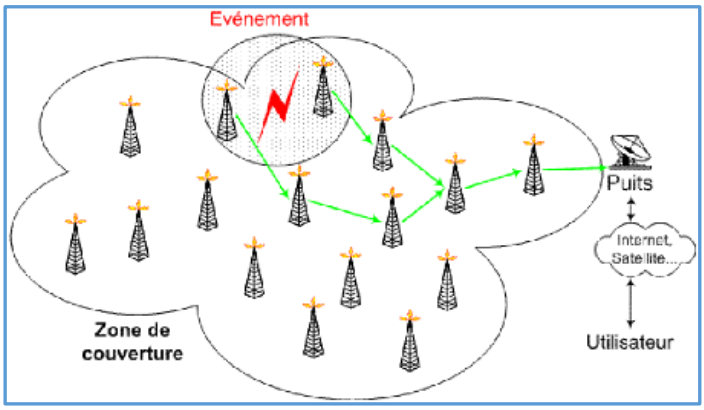

Figure 1.9 : collecte d'information 2

Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puit.

1.5.6 Anatomie d'un nœud capteur sans fil

A l'aide d'un capteur sans fil on peut mesurer plusieurs valeurs physiques environnementales (pression, température, lumière, etc) ainsi s'ouvre une communication avec le centre du contrôle via une station de base.
On rappelle que les capteurs sans fil fonctionnent comme des vrais systèmes embarqués avec la possibilité de traiter et de communiquer de l'information. Un nœud contient une architecture complètement dépendante de l'objectif de son déploiement [32]. Les réseaux de capteur sont composés de plusieurs unités, mais fondamentalement de quatre unités élémentaires :

- **Unité d'acquisition de données :** cette unité est divisée on deux sous-unités : les capteurs dont le rôle est de prendre des mesures numériques sur les paramètres environnementaux puis les transférer en signaux analogiques et à l'aide de la deuxième unité qui est dite ADCs ces signaux sont converties en numérique pour les interpréter par l'unité de traitement [32].
- **Unité de traitement des données :** Cette unité est composée de deux interfaces : l'une est l'unité d'acquittement, l'autre c'est module de transmission. Elle est capable de contrôler des procédures pour réaliser les taches d'acquittement à partir d'une collaboration entre les nœuds, et en même temps de stocker les données collectées [32].
- **Unité de transmission de données (Transceiver) :** A l'aide d'un transmetteur radio dont cette unité dispose les capteurs peuvent communiquer au sein d'un réseau. Au niveau de la réalisation de la transmission toutes les composantes sont classiques, de plus elles présentent les mêmes problèmes que les réseaux sans fil : l'accroissement de la distance impliquerait une augmentation au niveau de quantité d'énergie nécessaire à la transmission [27] et [32].
- **Unité d'alimentation** : Unité de puissance (Batterie) : A l'aide de cette unité qui prend une forme de batterie standard, toutes les autres unités citées précédemment vont disposer d'une source d'énergies et en même temps réduit les dépenses par une mise en veille des composantes inactives par exemple [27].

Il existe d'autres composantes d'un capteur sans fil comme un système de localisation pour identifier l'emplacement d'un capteur ainsi qu'un mobilisateur pour déplacer le capteur [29].

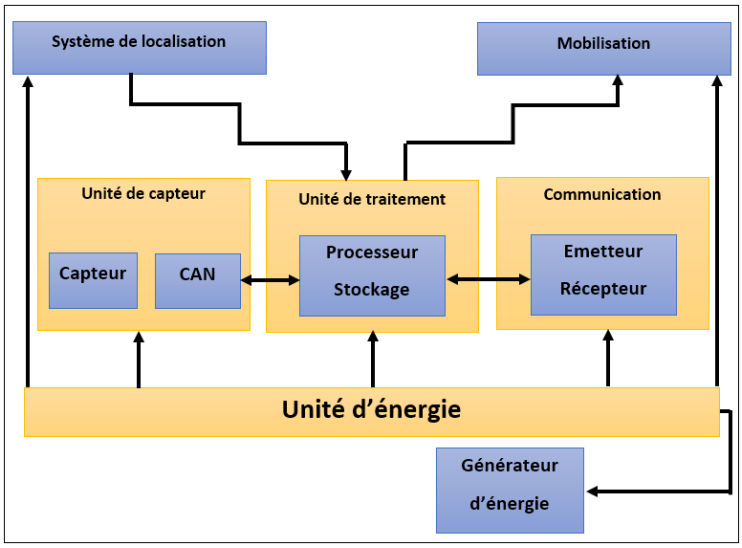

Figure 1.10 **:** Structure d'un capteur [29].

1.6 Domaine d'application :

Le domaine d'utilisation des capteurs sans fils est devenu de plus en plus vaste car ce nouveau genre de réseaux a attiré l'attention de l'industrie de l'informatique et des départements de recherches universitaires en raison de la popularité de nombreuses applications qu'elles peuvent effectuer, telles que la surveillance de l'environnement, les soins médicaux, la gestion des appareils électroménagers, la surveillance des champs de bataille et les scénarios de sécurité intérieure.

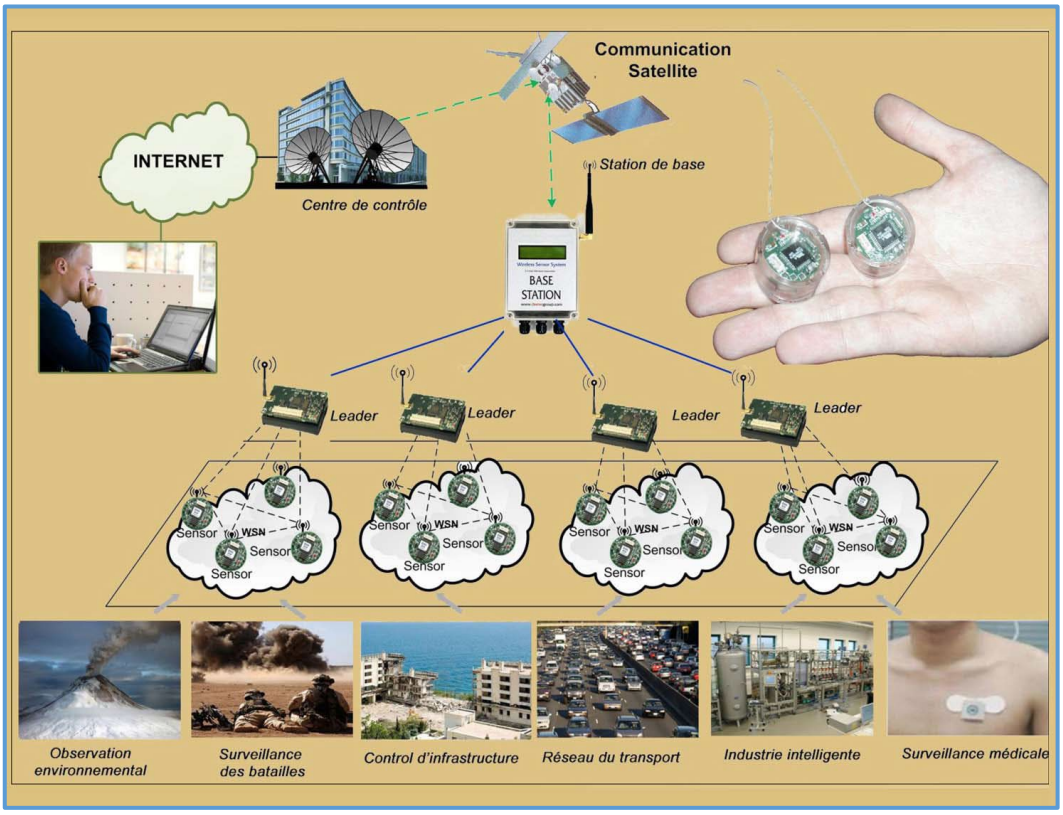

Figure 1.11 : application des RCSFs

La figure 1.11 illustre l'architecture d'un réseau de capteurs montre les vastes domaines qui appliquent ce type de réseau. La liste ci-dessous présente les distincts domaines qui exploitent

le principe des RCSFs :

- **Le domaine militaire :** Le réseau de capteurs sans fil a ajouté un impact très intéressant au niveau des applications militaires grâce à l'installation rapide sans aucune infrastructure précise. Ainsi, il offre des d'intérêts importants tels que la surveillance des mouvements de l'ennemi et la fiabilité de la communication à bas coût entre les unités avec une logistique peu compliquée.
- **L'environnement :** La petite taille, les capacités relativement grandes au niveau de calcul et de communication des capteurs permettent de les placer aux endroits que les humains ne peuvent ou ne veulent pas accéder, comme par exemple les grandes forêts, les volcans, les profondeurs des océans, les régions polaires, ou encore d'autres planètes que la terre. On peut aussi utiliser les RCSFs pour la surveillance du degré de maturité des récoltes (raisin), la mesure de la qualité de l'eau ou de l'air.
- **L'industrie :** Comme pour beaucoup d'autres domaines, les applications industrielles ont été les locomotives de la recherche pour les réseaux de capteurs. En fait, les industriels s'intéressent au potentiel des capteurs pour diminuer les coûts du contrôle et de la maintenance des produits, de la gestion de l'inventaire, de la télésurveillance après-vente, etc. En particulier, l'intégration de la technologie RFID (Identification par Radio-Fréquence) avec les réseaux de capteurs est une des directions prometteuses de recherche dans l'industrie.
- **Commercial :** Il est possible d'intégrer des nœuds de capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré.
- **Les domaines urbains et domotique :** Les capteurs entrent de plus en plus dans nos vies quotidiennes. Dans le milieu urbain, les capteurs sont déjà utilisés pour la localisation des bus, pour des tickets électroniques et pour la sécurité. Une des applications est la surveillance du trafic routier avec les réseaux de capteurs déployés sur les autoroutes. De plus, les maisons, les bâtiments, les bureaux équipés de capteurs intelligents permettent de construire des systèmes pervasifs où l'information est omniprésente.
- **Le domaine médical :** La recherche sur l'usage des capteurs intelligents dans le domaine médical inclut les moyens d'hospitalisation à domicile, l'intégration des microcapteurs "dans" le corps (e.g construire un BAN - Body Area Network) et la gestion des urgences. Parmi les applications les plus utiles, on cite la télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées ainsi que le contrôle à distance des données physiologiques. [32].

1.6.1 Les outils de simulation des RCSFs

Des simulateurs payants ou gratuits, dont certains sont listées ci-dessous, sont disponibles pour tester le comportement des réseaux de capteurs en modélisant chaque capteur, chaque station de base, et chaque protocole de communication, ensuite un trafic est généré dans le réseau simulé soit à partir des capteurs vers la station de base ou inversement :

- **NS2 (Network Simulator)** est l'outil le plus répandu et le plus utilisé. Il prote d'un grand support technique auprès de ses utilisateurs, cependant, ce simulateur souffre des erreurs

de dépassement de mémoire lors de la conception des grands réseaux de capteurs sans fil.

- **SensorSim** est une extension du simulateur NS2 pour les réseaux de capteurs sans fil. Il gère de façon dynamique le fonctionnement des capteurs, cependant, comme NS2, il souffre de dépassement de mémoire lors de l'utilisation des grands RCSFs.
- **SSFNet** est capable de gérer des réseaux de capteurs sans fil de grande taille avec un bon temps de traitement, cependant, les modèles des protocoles utilisés ne sont pas détaillés.
- **J-Sim** est un outil gratuit utilisant des scripts comme TCL, Phyton et Perl. Il peut traiter des réseaux de grandes tailles et comporte des librairies spécifiques pour les réseaux de capteurs sans fil, cependant, il définit une topologie fixe, ce qui limite son utilisation pour la conception de nouveaux protocoles.
- **SENSE** peut traiter des réseaux de grande taille, et gère bien la dépendance des modules et la réutilisabilité des composants, cependant, il n'est pas bien documenté et gère mal les interactions entre les composants internes.
- **TOSSIM** permet de simuler d'une manière évolutive les réseaux de capteurs sans fil, cependant il ne fonctionne qu'avec le système d'exploitation "TinyOS" installé dans un capteur.

1.7 Les avantages et les contraintes d'un RCSF

Les réseaux de capteurs sans fil feront dans les années à venir une partie intégrante de notre vie quotidienne et changeront certainement notre manière de vivre. Cependant, comme tous les types de réseaux, ce réseau porte beaucoup d'avantages mais aussi des limites que l'utilisateur prend en considération lors de déploiement.

- **Les avantages :**
  - Coût de plus en plus faible
  - Taille réduite
  - Une large gamme
  - Facilité de déploiement
  - Auto-organisation des capteurs
  - Support de communication sans fil
  - Tolérance aux pannes
- **Les Contraintes** :

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit :

- ***La tolérance de fautes*** **:** Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces problèmes n'affectent pas le reste du réseau, c'est le principe de la tolérance de fautes qui correspond à la capacité de maintenir les fonctionnalités du réseau sans interruptions dues à une erreur intervenue sur un ou plusieurs capteurs.
- ***L'échelle :*** Le nombre de nœuds déployés pour un projet peut atteindre le million. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que le puits "sink " soit équipé de beaucoup de mémoire pour stocker les informations reçues.

- ***Les coûts de production :*** Souvent, les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un nœud est critique afin de pouvoir concurrencer un réseau de surveillance traditionnel. Actuellement un nœud ne coûte souvent pas beaucoup plus que 1$. A titre de comparaison, un nœud Bluetooth, pourtant déjà connu pour être un système low-cost, revient environ à 10$.
- ***L'environnement :*** Les capteurs sont souvent déployés en masse dans des endroits 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, Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées.
- ***La topologie de réseau :*** Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : Déploiement, Post-déploiement (les capteurs peuvent bouger, ne plus fonctionner), Redéploiement de nœuds additionnels.
- ***Les contraintes matérielles :*** La principale contrainte matérielle est la taille du capteur. Les autres contraintes sont que la consommation d'énergie doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (fortes chaleurs, eau,..), qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.
- ***Les médias de transmission :*** Dans un réseau de capteurs, les nœuds sont reliés par une architecture sans-fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être normé. On utilise le plus souvent l'infrarouge (qui est license-free, robuste aux interférences, et peu onéreux), le bluetooth et les communications radio ZigBee.
- ***La consommation d'énergie :*** Un capteur, de par sa taille, est limité en énergie (< 1.2V). Dans la plupart des cas le remplacement de la batterie est impossible. Ce qui veut dire que la durée de vie d'un capteur dépend grandement de la durée de vie de la batterie. Dans un réseau de capteurs (multi-sauts) chaque nœud collecte des données et envoie/transmet des valeurs. Le dysfonctionnement de quelques nœuds nécessite un changement de la topologie du réseau et un re-routage des paquets. Toutes ces opérations sont gourmandes en énergie, c'est pour cette raison que les recherches actuelles se concentrent principalement sur les moyens de réduire cette consommation.

1.8 La sécurité des réseaux capteurs

Mettre en œuvre une politique de sécurité contre les attaquants qui veulent perturber ou détruire un RCSF est très difficile à cause des ressources limitées de ces capteurs. Donc, cette section a comme rôle de classifier les objectifs, les défis ainsi les problèmes majeurs de la sécurité.

1.8.1 Objectifs de la sécurité dans les RCSFs

La connaissance des objets à protéger est le premier pas qu'on doit suivre dans le but de déterminer les différents objectifs de sécurité. Parmi eux nous citons :

- **Disponibilité du réseau :** Le mot « Disponibilité » reflète l'existence d'un réseau pour assurer ses services et autoriser les parties communicantes lorsque ceci est nécessaire [32]. A comparaison des contraintes de ce type des réseaux, cet objectif reste difficile à appliquer à cause de changement de la topologie, ainsi que les communications sans fil

impliquent la facilité d'avoir une liaison brouillée et perturbée entre les nœuds de capteurs. Ceci garantit aux adversaires de lancer des attaques de déni de service (Dos) ciblant la performance du réseau ou même détruisant l'ensemble du réseau. [33].

- **Authentification :** Elle permet de coopérer au sein des RCSFs sans risque, en contrôlant et en identifiant les participants. Elle apparaît comme la pierre angulaire d'un réseau de capteur sans fil sécurisé. En effet, on ne peut assurer une confidentialité et une intégrité des messages échangés si, dès le départ, on n'est pas sûr de communiquer avec le bon nœud. Si l'authentification est mal gérée, un attaquant peut se joindre au réseau et injecter des messages erronés. L'utilisation de Code d'Authentification de Message (CAM), ou MAC en anglais (Message Authentification Code), permet d'assurer à la fois l'authentification de l'origine et l'intégrité du message. Un exemple de MAC est : HMAC (keyed-Hash Message Authentification Code).

- **Intégrité des données :** Cette propriété exige d'avoir une ressemblance entre les messages émis et reçus c'est à dire que les données qui arrivent au destinataire ne doivent pas être altérées durant leur transmission dans le réseau de manière volontaire ou accidentelle. Elle peut être assurée par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique. Les fonctions MD2 (Message Digest 2), MD5, SHA-1 (Secure Hash Algorithm 1) sont des exemples de quelques fonctions de hash les plus utilisées.

- **La confidentialité :** Une fois les parties authentifiées, la confidentialité reste un point important, étant donné la communication sans fil des RCSFs. Elle consiste à préserver le secret des messages échangés et ne pas les révéler aux adversaires. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique.

- **Auto organisation :** Cet objectif peut être nécessaire dans plusieurs cas. Par exemple, un réseau comportant un grand nombre de nœuds, placés dans des endroits hostiles où la configuration manuelle n'est pas faisable, doit être capable de s'auto organiser. Un autre cas est celui où un nœud est inséré ou retiré (à cause d'un manque d'énergie ou de destruction physique), ainsi le réseau doit être capable de se reconfigurer pour continuer sa fonction. [34].

- **Fraîcheur des données :** Il implique que les données sont récentes, et que l'attaquant n'a pas retransmis d'anciens messages. Pour résoudre ce problème, un numéro de séquence peut être ajouté aux paquets de données pour filtrer les anciens messages. [34] **Comme *le principe de traitement à l'intérieur du réseau « in-network processing »***

- **Localisation sécurisée :** La localisation est un facteur très important pour la fiabilité de fonctionnement des RCSFs. En effet, un réseau de capteurs doit être capable de localiser automatiquement chaque capteur dans le réseau. Ainsi, un réseau de capteurs conçu pour localiser des événements aura besoin d'informations précises sur la localisation afin de repérer la position exacte de ces derniers.

  Un nœud malveillant peut essayer de compromettre les informations de localisation afin de déstabiliser le fonctionnement du réseau, ce qui rend la sécurité de localisation un objectif très important pour les systèmes de sécurité. [35].

1.8.2 Issues majeures de la sécurité

La sécurité est un domaine très vaste et représente un défi scientifique à cause des caractéristiques spécifiques des réseaux de capteurs, Nous citons quelques exemples de mécanismes de sécurité dédiés aux réseaux de capteurs :

- **La sécurité du routage :** Le problème du routage consiste à déterminer un

acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance comme la consommation d'énergie. Une attaque simple de déni de service sur un protocole de routage consiste pour un nœud à refuser arbitrairement de transférer certains messages ou de supprimer un paquet et le déplacer de façon aléatoire. L'attaque du trou de ver peut également faire croire à deux nœuds distants qu'ils sont très proches alors qu'en réalité ils sont éloignés de plusieurs sauts. Les nœuds du réseau seront alors contraints de mettre à jour leur table de routage pour continuer d'assurer la fiabilité de leur service.

Il est nécessaire de faire une sécurité sur les protocoles de routage conçus initialement pour un environnement sans risque et aussi de concevoir des algorithmes robustes afin de mener à bien l'opération de l'acheminement des données même en présence des nœuds malicieux.

- **La sécurité de l'agrégation de données :** Une approche courante pour maîtriser les limitations des réseaux de capteurs est d'agréger les données au niveau des nœuds intermédiaires. Garantir la sécurité en même temps avec des techniques d'agrégation est difficile parce qu'un nœud capturé pose un double problème. Il compromet la confidentialité des données (possibilité d'écoute) et leur disponibilité (possibilité d'attaque du type déni de service). Egalement, un nœud d'agrégation compromis met en danger toutes les mesures qui font partie de l'agrégat dont le nœud est responsable ce qui mène à déclencher de fausses alarmes ou même de dissimuler les événements d'exception dans les applications critiques (*on aura plus de détails sur cette approche dans le chapitre 2*).

- **La sécurité de la localisation :** La connaissance des positions des capteurs dans l'environnement surveillé ils sont indispensable pour une grande majorité des applications et aussi de déterminer l'origine des événements détectés. Et pour utiliser aussi la localisation pour les protocoles de routage géographique dans les réseaux à grande échelle, il est nécessaire de faire la localisation pour faire la transmission des données seulement dans la direction de la destination avec la meilleure précision possible pour tous les nœuds du réseau. Cependant la plupart des capteurs ne peuvent être dotés d'un récepteur GPS et dépendent d'un certain type de capteurs nommés ancres pour estimer leur position. Les nœuds de capteurs sont toujours éloignés de plusieurs sauts d'ancres et calculent leurs coordonnées sur la base des coordonnées des ancres et le délai qui les séparent. Donc la sécurisation des protocoles de localisation est nécessaire pour protéger le réseau des ancres malicieux et des attaquants qui tentent de troubler le processus de localisation.

- **La gestion de clés :** Dans le but de fournir les services de sécurité tels que : confidentialité, authentification, intégrité, sécurité de routage/agrégation/localisation etc., les capteurs ont besoin de partager et d'établir un nombre de clés cryptographiques secrètes. Ceci est effectué grâce à la gestion de clés qui sont des mécanismes efficaces, sécurisés et stables de gestion de clés utilisées dans les opérations cryptographiques. Donc la gestion de clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Les nœuds capteurs sont potentiellement exposés aux attaques physiques et ne peuvent compter sur une intervention humaine. Un attaquant qui capture un nœud peut extraire toutes ses clés secrètes et déclencher tout type d'attaques sans qu'il soit identifié. Donc, les protocoles de gestion de clés doivent être résistants aux attaques contre les capteurs. Une astuce de distribution sécurisée des clés est également à prévoir afin de pouvoir assurer un certain niveau de sécurité. [32]

1.8.3 Défis de la sécurité dans RCSFs

Les Réseaux de capteur ayant plusieurs caractéristiques qui les rendent très vulnérables à des attaques malicieuses dans les différents environnements. Par exemple, les attaques de déni de service, l'injection et la modification des données, la retransmission sélective, etc. peuvent être dangereuses pour le RCSF, ce qui rend difficile d'appliquer directement les approches de sécurité existantes sur des réseaux de capteurs sans fil. Par conséquent, afin d'élaborer des mécanismes de sécurité utiles tout en empruntant des idées des techniques de sécurité actuelles, il est nécessaire de comprendre les caractéristiques suivantes :

- **La contrainte des ressources :** Toutes les approches de sécurité nécessitent une certaine quantité de ressources pour leur mise en œuvre, y compris la mémoire de données, l'espace du code et l'énergie pour alimenter le capteur. Toutefois, ces ressources sont très limitées dans ces petits appareils de capteurs sans fil en raison de la prise en considération des coûts de fabrication :
    - **La mémoire limitée et l'espace de stockage :** Un capteur est un petit appareil avec seulement une petite quantité de mémoire et d'espace de stockage pour le code. Afin de construire un mécanisme de sécurité efficace, il est nécessaire de limiter la taille du code de l'algorithme de sécurité. En moyenne, la plupart des nœuds de capteurs sans fil possèdent un microcontrôleur de 8-16 bits, avec seulement 10-64K de mémoire programme et 512K-4Mo capacité de stockage flash.
    - **Contrainte de la puissance :** L'énergie utilisée est le plus grand obstacle à la capacité des capteurs sans fil. Nous supposons que, une fois que les nœuds de capteurs sans fil sont déployés dans un réseau, ils ne peuvent plus être facilement remplacés ou rechargés. Par conséquent, l'énergie initiale des capteurs doit être conservée pour prolonger leur durée de vie individuelle et la durée de vie du réseau de capteurs en général. Lors de l'implémentation des protocoles de sécurité dans les capteurs, l'impact énergétique du code de sécurité ajouté doit être considéré.

      Lors de l'ajout de sécurité à un nœud de capteur, nous nous intéressons à l'impact de la sécurité sur la durée de vie d'un capteur (par exemple, la durée de vie de sa batterie). La puissance supplémentaire consommée par les nœuds de capteurs en raison de la sécurité est liée au traitement requis pour les fonctions de sécurité (le chiffrement, le déchiffrement, la signature de données, la vérification des signatures), l'énergie nécessaire pour transmettre les données relatives à la sécurité (par exemple: les vecteurs d'initialisation nécessaires pour le chiffrement/déchiffrement) et l'énergie nécessaire pour sauvegarder les paramètres de sécurité d'une manière sûre (par exemple, le stockage de clefs de chiffrement).

      Lors de la conception de la plupart des protocoles de sécurité dans les RCSFs, le chiffrement (cryptage) à clef privée (par exemple, DES, RC5) est préférable au chiffrement à clef publique (Diffie-Hellman, RSA). Ce dernier est possible, mais consomme beaucoup plus de ressources.

      Un RCSF formé de milliers de capteurs nécessite des protocoles de sécurité simples et flexibles. Par contre, construire ce genre de protocoles n'est pas une tâche facile. Un protocole de sécurité plus robuste utilise plus de ressources au niveau des capteurs, ce qui peut conduire à une dégradation de performance au niveau des applications. Dans la plupart des cas, un compromis entre la sécurité et la performance doit être effectué. Évidemment, les faibles protocoles de

sécurité peuvent être plus faciles à attaquer par les adversaires.

- **La fiabilité des communications :** Un canal sans fil est un moyen de communication ouvert qui peut être consulté par toute personne dans la portée du signal. Cependant, cette ouverture démontre un grand avantage, car elle réduit le coût de l'infrastructure.

  Mais avec une interface radio configurée à la même bande de fréquence, les adversaires peuvent surveiller ou participer à la communication. Ceci pose plusieurs problèmes qui sont expliqués ci-dessous :

  - **Manque de fiabilité de transfert :** Contrairement aux réseaux de capteurs filaires (connectés avec des fils), les canaux sans fil ne sont pas aussi fiables.

    Ils sont sensibles aux interférences, aux erreurs du canal, à la congestion (requêtes supérieures à la capacité du traitement dans les capteurs) et aux différents objets se déplaçant dans la portée du signal. Ces conditions peuvent être permanentes ou temporaires et peuvent endommager ou laisser tomber des paquets (un rapport demande un ou plusieurs paquets) sur le réseau sans fil. Si un protocole sans fil ne prévoit pas la gestion des erreurs, il peut conduire à une communication incohérente et à la perte des paquets critiques de sécurité (par exemple, une clef de chiffrement), produisant ainsi des nœuds de capteurs incapables de communiquer de façon sécurisée.

  - **Conflits/Collisions :** Même si nous supposons que le canal est fiable, la communication peut encore ne pas être fiable à cause des collisions de paquets dans le canal sans fil. Cela est dû à la nature d'émission dans les réseaux de capteurs sans fil. Si deux capteurs voisins (l'un à la portée du signal de l'autre), tentent d'émettre un paquet chacun en même temps, une collision entre les deux paquets peut se produire et le transfert lui-même peut échouer. Dans un réseau de capteurs de grande densité, cela peut être un problème majeur.

  - **Temps de latence :** La synchronisation entre les nœuds de capteurs sera une tâche difficile et affectera la déclaration des événements et la distribution de clefs cryptographiques. Ceci est d'autant plus vrai lorsque le routage à partir des liaisons multiples (aussi appelé multi-hop), la congestion du réseau et le traitement des données au niveau des nœuds de capteurs augmentent le délai normal de transmission dans le réseau.

- **Le fonctionnement autonome :** Un des principaux avantages des réseaux de capteurs est la possibilité de placer les nœuds de capteurs dans un environnement sans aucune surveillance. Ceci peut produire des faiblesses de sécurité pour le réseau si les nœuds de capteurs sont déposés dans des environnements difficiles ou d'une manière non garantie. L'absence de la protection physique peut permettre à plusieurs types d'attaques.

  - **L'exposition à l'environnement ou Attaques physiques :** Les nœuds de capteurs sans fil peuvent être déployés dans un environnement ouvert à des agressions physiques. Par exemple, les nœuds de capteurs dans l'océan peuvent être mangés par les poissons ou emportés pendant les orages. Étant donné que ces nœuds sont déployés dans des environnements ouverts, ils peuvent également être attaqués ou volés par des adversaires.

  - **Gestion à distance :** Un des avantages des réseaux de capteurs sans fil est leur capacité d'être gérés à distance. Ceci permet aux nœuds de capteurs d'être placés dans des environnements dangereux ou inaccessibles. Par contre la gestion à distance exige la présence des mécanismes de sécurité pour protéger le réseau et les informations transmises au centre de contrôle. La sécurité est également

nécessaire pour protéger les serveurs du centre de contrôle étant donné que le réseau peut être utilisé par des adversaires afin d'accéder aux systèmes du serveur principal.

- **Infrastructure non-fixe :** Les réseaux de capteurs peuvent s'organiser automatiquement pour former un réseau distribué. Cela fournit un réseau de communication robuste et dynamique pour envoyer des informations aux serveurs dans le monde extérieur. Par contre, si un réseau est mal conçu, l'organisation de ce réseau devient difficile, inefficace et fragile. La communication entre les nœuds de capteurs nécessite d'intégrer des fonctionnalités de sécurité contre les attaques possibles. [33]

1.9 Conclusion

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement, industrie et même dans le domaine sportif. Dans ce premier chapitre, nous avons présenté les RCSFs, leurs spécificités et les concepts nécessaires à la compréhension des réseaux de capteurs. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux ainsi que les notions de sécurité essentielles pour effectuer un réseau fiable et sécurisé.

Chapitre

2

Les attaques sur les réseaux de capteurs sans fil

2.1 Introduction

Tout capteur connecté à un réseau informatique est potentiellement vulnérable à des attaques. En effet, une « attaque » [36] est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables. Nous présentons dans ce chapitre les natures, les origines et aussi les différents types d'attaques qui peuvent perturber ou détruire un réseau de capteur sans fil.

2.2 Natures des attaques

Les attaques peuvent à première vue être classées en 2 grandes catégories :

On parlera d'attaque active si un attaquant modifie l'état du réseau, et d'attaque passive dans le cas où il ne cherchera qu'à l'écouter.

2.2.1 Attaque passive

Dans un réseau sans fil l'écoute passive est d'autant plus facile que le média air est difficilement maîtrisable. Bien souvent, la zone de couverture radio d'un point d'accès déborde du domaine privé d'une entreprise ou d'un particulier.

L'attaque passive la plus répandue est la recherche de point d'accès. Cette attaque (appelée Wardriving) est devenue le "jeu" favori de nombreux pirates informatique, les points d'accès sont facilement détectables grâce à un scanner (portable équipé d'une carte WIFI et d'un logiciel spécifique de recherche de PA (Point Acces)). Ces cartes wifi sont équipées d'antennes directives permettant d'écouter le trafic radio à distance hors de la zone de couverture du point d'accès. Il existe deux types de scanners, les passifs (Kismet, Wifiscanner, Prismstumbler…) ne laissant pas de traces (signatures), quasiment indétectables et les actifs (Netstumbler, dstumbler) détectables en cas d'écoute, ils envoient des "probe request". Seul Netstumbler fonctionne sous Windows, les autres fonctionnent sous Linux. Les sites détectés sont ensuite indiqués par un marquage extérieur (à la craie) suivant un code (warchalking) : Une première analyse du trafic permet de trouver le SSID (nom du réseau), l'adresse MAC du point d'accès, le débit, l'utilisation du cryptage WEP (Wired Equivalent Privacy) et la qualité du signal. Associé à un GPS, ces logiciels permettent de localiser (latitude longitude) ces points d'accès. A un niveau supérieur des logiciels (type Aisnort ou Wepcrack) permettent, en quelques heures (suivant le trafic), de déchiffrer les clés WEP et ainsi avec des outils d'analyse de réseaux conventionnels la recherche d'information peut aller plus loin. Le pirate peut passer à une attaque dite active [36].

2.2.2 Attaque active

Ce genre d'attaque se concentre sur la modification ou la suppression des données ou des

messages, il peut être perturbateur majeur de bon fonctionnement de réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

- **DoS (Denial of Service) :** Les attaques par déni de services (denial of service ou DoS) sont des attaques qui visent à rendre un réseau indisponible durant une certaine période. Cette attaque a pour but d'empêcher des utilisateurs légitimes d'accéder à des services en saturant de fausses requêtes destinées à ces types des services. Elle se base généralement sur des " bugs " logiciel. Dans le milieu wifi, cela consiste notamment à bloquer des points d'accès soit en l'inondant de requête de désassociassions ou de désauthentification (programme de type Airjack), ou plus simplement en brouillant les signaux hertziens.

- **Spoofing (usurpation d'identité) :** Le spoofing IP est une technique permettant à un pirate d'envoyer à une machine des paquets semblant provenir d'une adresse IP autre que celle de la machine du pirate. Le spoofing IP n'est pas pour autant un changement d'adresse IP. Plus exactement il s'agit d'une mascarade (il s'agit du terme technique) de l'adresse IP au niveau des paquets émis, c'est-à-dire que les paquets envoyés sont modifiés afin qu'ils semblent parvenir d'une machine légitime.

- **Man in the middle (homme au milieu) en milieu Wi-Fi :** Cette attaque consiste, pour un réseau Wi-Fi, à disposer un point d'accès étranger dans à proximité des autres PA légitimes. Les stations désirant se connecter au réseau livreront au PA leurs informations nécessaires à la connexion. Ces informations pourront être utilisées par une station pirate. Il suffit tout simplement à une station pirate écoutant le trafic, de récupérer l'adresse MAC d'une station légitime et de son PA, et de s'intercaler au milieu [36].

2.3 Ses origines (interne /externe)

2.3.1 Interne

Un attaquant interne est celui qui arrive à contrôler un nœud ayant le statut de membre à part entière du réseau et qui dispose donc de l'ensemble des connaissances associées à ce statut (clés secrètes, table de routage, etc..). Il est considéré comme la plus dangereuse du point de vue sécurité. Puisque l'attaquant qui capture un nœud, et de lire sa mémoire et accéder à son matériel cryptographique et par conséquent peut s'authentifier comme un nœud légitime et émettre des messages aléatoires erronés sans qu'il soit identifié comme intrus, puisqu'il utilise des clés valides. Les méthodes cryptographiques s'avèrent donc inefficaces pour ce genre d'attaque. Il est donc nécessaire d'utiliser d'autres méthodes complémentaires telles que les systèmes de monitoring et les systèmes de réputations.

2.3.2 Externe

Un nœud attaquant externe ne dispose pas a priori des connaissances sur le réseau ciblé. En particulier, il est incapable d'effectuer des opérations cryptographiques comme signer, déchiffrer, etc. L'utilisation de systèmes de chiffrement et de signature pour sécuriser les communications est un moyen efficace de restreindre les actions d'un attaquant externe. En effet, dans un tel cadre, les actions disponibles pour l'attaquant sont de provoquer une congestion dans le réseau, déni de services (DoS), et d'injecter de fausses informations de routage pour ensuite rendre le système inutilisable. Les attaques externes empêchent la communication normale du réseau et génèrent un trafic supplémentaire au réseau.

2.4 Les différents types d'attaques

Il existe plusieurs types d'attaques qui ont pour but de perturber ou de détruire complétement le réseau de capteurs sans fil, dans cette section nous citons quelques exemples d'attaques :

2.4.1 Forced Delay

Un nœud malveillant retarde délibérément les paquets à l'intérieur de son élément de transmission afin de retarder la transmission des événements importants. Cette attaque peut être efficacement utilisée pour dégrader la qualité de service des applications dans les systèmes en temps réel [37].

2.4.2 Sybil

« Sybil » provient du titre du roman éponyme écrit par Flora Rheta Schreiber, publié en 1973 et raconte le traitement d'une patiente souffrant de multiples dédoublements de la personnalité [38].

C'est une attaque de l'identité multiples. Son principe exige qu'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.

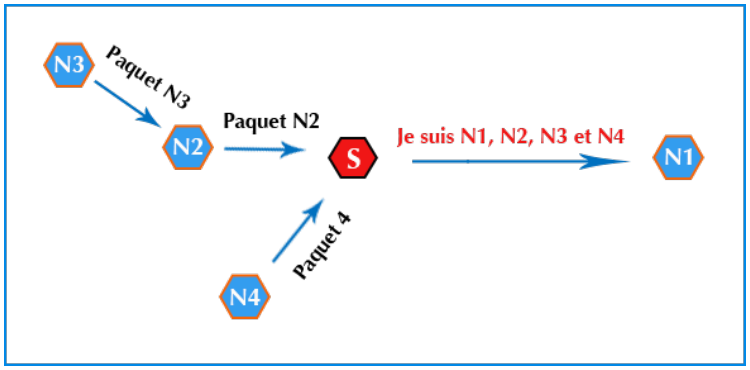

Figure 2.1 Fonctionnement d'attaque Sybil

Cette attaque est capable de réaliser plusieurs fonctionnalités qui touchent le matériel ainsi que le routage des données échangées sur un RCSF.

- **Mémorisation distribuée :** un même nœud malveillant peut être capable de stocker autant de données qu'il prétend avoir d'identités différentes.
  Étant donné que ce nœud malveillant représente un unique point de défaillance, il peut ainsi facilement faire les mécanismes de fragmentation et de réplication.
- **Routage :** un même nœud malveillant peut être capable de relayer autant de paquets qu'il présente d'identités Sybille. Au lieu de router ces paquets, il peut les supprimer, ou modifier les données qu'ils contiennent.
- **Agrégation de données :** un nœud malveillant peut affecter l'agrégat calculé étant donné qu'il peut contribuer autant de fois qu'il prétend avoir d'identités différentes.
- **Vote :** un nœud malveillant peut être capable de déterminer le résultat de n'importe quel vote en faisant voter toutes ses identités Sybille pour une même entité.
- **Partage équitable de ressources :** une attaque Sybille peut être mené dans le but de

permettre à un seul nœud de bénéficier d'une répartition inéquitable des ressources partagées avec d'autres nœuds. [39] Les techniques d'authentification et de chiffrement peuvent empêcher un étranger de lancer une attaque Sybil sur le réseau de capteur. [38]

2.4.3 Sinkhole

L'attaque Sinkhole se produit lorsqu'un nœud essaye d'attirer tout le trafic. Ceci est possible lorsque l'attaquant diffuse des valeurs très intéressantes à ses voisins en ce qui concerne le choix des métriques de routage. Ainsi, les nouds voisins vont choisir le nœud malicieux comme prochain saut. Par exemple, la figure ci-dessous montre que le nœud noir (malveillant) se comporte comme un voisin de la station de base, qui est le plus souvent le point qui recueille le plus d'informations de l'intégralité du réseau pour attirer tout le trafic. Et aussi le nœud malicieux va proposer aux nœuds le chemin le plus rapide pour atteindre la base, en utilisant une connexion plus puissante. Ainsi l'ensemble de ces nœuds va s'adresser en particulier à ce nœud malicieux pour transmettre l'information à la base. Toutes les informations qui transitent de ces nœuds vers la base pourront être récupérées par l'attaquant. Pour générer une attaque encore plus puissante, un attaquant peut utiliser des stratégies de type trou de ver associées à une attaque de type trou de la base.

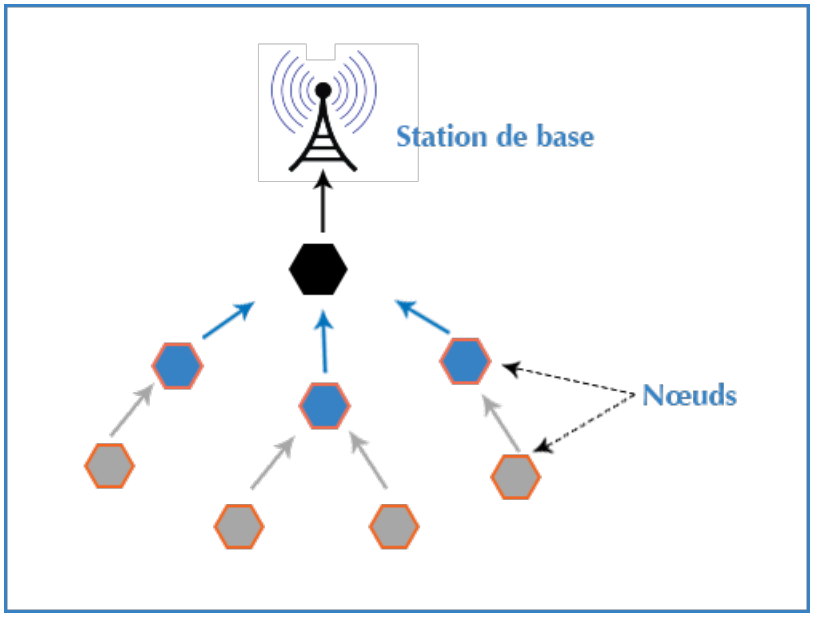

Figure 2.2 : Schéma d'attaque Sinkhole

Cette situation est représentée sur la figure ci-dessous, où les nœuds malicieux X_1, X_2 et X_3 sont reliés par des connexions puissantes et forment des trous de ver. X_3 est lui relié à la base par une connexion puissante pour réaliser une attaque du trou de la base. On parle alors d'une sphère d'influence exercée par l'attaquant sur le réseau, car il est ainsi capable de récupérer l'intégralité des informations qui circulent dans le réseau de capteurs sans fil.

L'intention de l'attaque Sinkhole est de créer une fausse station de base et ainsi une fausse topologie. Elle nécessite des ressources plus élevées pour convaincre les nœuds de sa supériorité. Lancée généralement par un ordinateur portable ou un PDA, elle exploite le fait que les identités et les liens entre les nœuds ne sont pas vérifiés. Réalisée au niveau de la couche de routage, cette attaque a pour cible tous les services fournis par le réseau. Le résultat est que les informations n'atteignent pas la station de base, causant ainsi des dommages partiels/totaux au réseau.

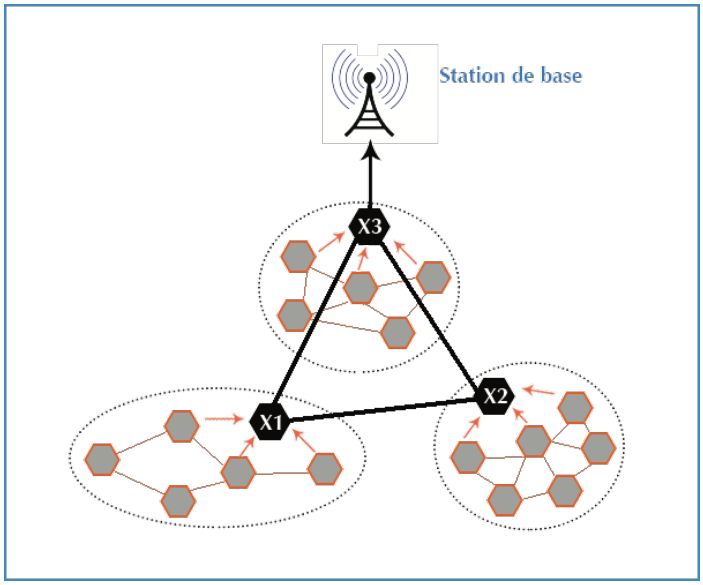

Figure 2. 3 : Schéma d'attaque Sinkhole

2.4.4 Wormhole

L'attaque du trou de ver est une attaque particulièrement difficile à contrer. Elle peut être lancée par un attaquant externe et être réussie même en présence d'un système d'authentification et de chiffrement.

Une attaque Wormhole consiste à créer un lien ou un tunnel de faible latence entre deux nœuds malicieux dans le réseau. L'attaque Wormholes (figure 2.4) est une attaque critique, où l'attaquant énumère les paquets d'une localisation, et les transporte à une autre place. Cette attaque n'a pas besoin de compromettre un capteur du réseau de capteurs sans fil, et peut même s'exécuter à la phase de découverte des capteurs voisins. Les attaquants peuvent coopérer afin de fournir une basse latence pour les communications.

Ainsi, quand les attaquants cessent de véhiculer leurs messages, l'état du réseau de capteurs devient instable, et requiert une réinitialisation.

Le renvoi géographique est la défense adéquate, qui résiste à ces attaques, car chaque message est envoyé au capteur le plus proche physiquement. Chaque message a un horodatage et une localisation de son émetteur. Le récepteur compare ces informations avec sa propre localisation et horodatage pour vérifier si les intervalles de transmissions sont dépassés.

Cette figure montre une situation où une attaque "Wormhole" prend émet. Quand un capteur B (une station de base ou un capteur intermédiaire) envoie un paquet de routage, l'attaquant reçoit ce paquet et le renvoie à ses capteurs voisins Z. Chaque voisin, recevant ce paquet renvoyé, se considéré comme voisin du capteur B, et le marque comme son parent. Par conséquent, même si le capteur victime Z est à plusieurs sauts du capteur B, l'attaquant lui transmet l'information qu'il est à un saut seulement du capteur B, créant ainsi un "Wormhole".

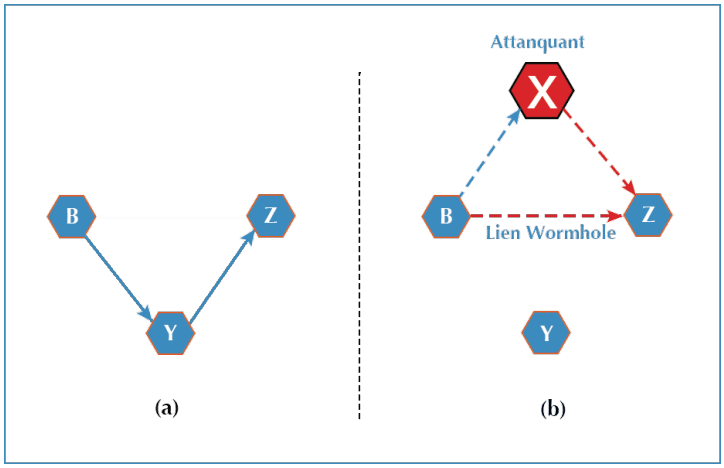

Figure 2.4 : Schéma présente l'attaque Wormhole

2.5 Les attaques connues et leurs solutions

Une variété d'attaques contre les RCSFs est rapportée dans la littérature. Pour faire face à ces attaques, diverses contre-mesures ont été proposées. Nous présentons par la suite les principaux types d'attaques, leurs fonctionnements et leurs solutions de défense.

2.5.1 Jamming

Son principe consiste à empêcher les capteurs d'un réseau sans fil de communiquer normalement. Elle se situe au niveau de la couche physique du modèle réservé pour la communication des nœuds de capteurs. [39] Pour effectuer ce processus, il faut qu'un adversaire identifie les fréquences radio utilisées par le RCSF visé et essaie de perturber ou de bloquer ces communications. Il s'agit de mettre une antenne pour émettre des signaux sur les mêmes fréquences afin d'empêcher les nœuds de communiquer sur ces fréquences. D'abord, l'adversaire place son antenne proche des nœuds ciblés par l'attaque. La plupart des adversaires préfèrent attaquer une station qui représente un point de passage obligatoire pour une grande partie du trafic comme un routeur ou une station de base.

2.5.2 Les différentes stratégies

Il y a quatre stratégies utilisées par l'attaque Jamming qui sont :

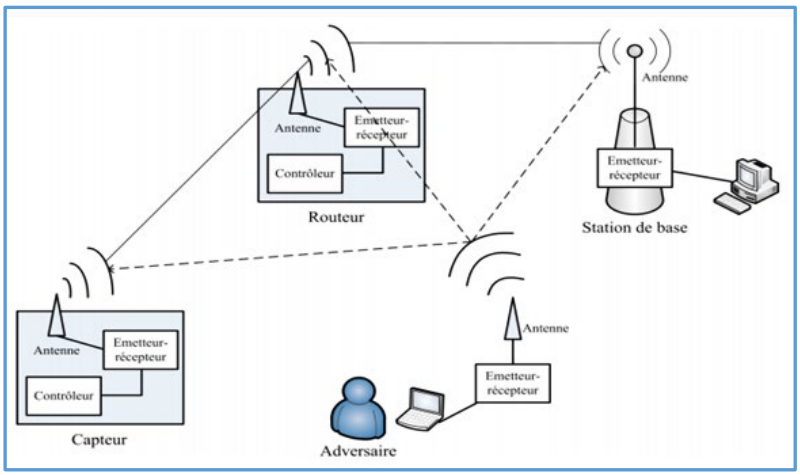

Figure 2.5 : Principe d'attaque Jamming

2.5.2.1 Le Jamming constant corrompt les paquets en transmission dans le RCSF, mais en contrepartie, l'attaquant doit avoir plus de ressources que ses victimes.

2.5.2.2 Le Jamming trompeur envoie une trame constante dans le réseau, par exemple dans tinyOS, le dispositif reçoit des bits constants, ce qui oblige les capteurs du réseau à rester en mode réception, ainsi ils ne peuvent plus renvoyer des données sur le réseau.

2.5.2.3 Le Jamming aléatoire alterne entre l'état de veille et l'état de Jamming, afin d'économiser de l'énergie.

2.5.2.4 Le Jamming réactif transmet seulement un signal de Jamming en cas de détection d'un signal sur le réseau, mais l'identification du Jamming réactif est difficile car il peut être remarqué comme un paquet en collision. [29]

2.5.3 Méthodes de défense

Il existe plusieurs techniques de défense contre le Jamming sur la station de base :

- **La réplication de la station de base :** La station de base remplaçante pourra prendre le relai si elle n'est pas attaquée elle aussi.
- **Le changement de place de la station de base :** Cette technique permet de varier le déplacement de la station de base, ce qui lui permet de s'éloigner de la source de l'attaque pour se trouver hors de portée du Jamming.
- **Le changement du chemin des données (technique multipath) :** Dès qu'un nœud détecte un Jamming sur le chemin vers la station de base, il tente d'envoyer les données via un autre chemin.

Dans l'étude présenté de la Ref. [40], les auteurs proposent une technique hybride qui réunit les trois techniques présentées ci-dessus afin de se défendre des effets du Jamming sur une station de base. Une technique qui peut être utilisée par tous les nœuds, est le saut de fréquences ou frequency hopping spread spectrum (FHSS). Le but est de permettre aux nœuds de changer de fréquence en suivant une séquence pseudo aléatoire connue par l'expéditeur et le destinataire (ou l'ensemble des destinataires). Pour attaquer le réseau dans ce cas, il faudra que l'adversaire arrive à occuper une partie importante des fréquences utilisées dans la séquence de sauts. Notons qu'il existe des moyens simples qui permettent de limiter le déni-de-service causé par le Jamming.

Par exemple, si un nœud reçoit en permanence des signaux et arrive à identifier que c'est un Jamming, il pourra alors les ignorer ou se mettre en mode veille pour un moment (ce qui provoquera une réaction au niveau du routage des paquets). Il se réveillera de temps en temps pour voir si l'attaque est toujours active. [41]

On pourrait aussi contrer ce type d'attaque au niveau des nœuds par commutation rapide des paquets de données sur plusieurs fréquences (couteuse et complexe). Par isolation de la zone infectée afin de contourner les nœuds malicieux.

2.5.4 Hello Flood

L'attaque « Hello flood » est effectuée par un attaquant disposant de grandes ressources, qui envoie des messages « HELLO » à un grand nombre de capteurs, dans une large région de réseau de capteurs sans fil. Ainsi, les capteurs victimes croient que les adversaires sont leurs voisins, et leurs envoient des messages qui devraient aboutir à la station de base. Le nœud malicieux diffuse les paquets Hello et génère un signal assez puissant comparativement aux

autres nœuds. Dans ce cas, d'autres nœuds légitimes envoient leurs paquets vers ce nœud malicieux. En conséquence, les paquets seront ensuite supprimés ou modifiés.

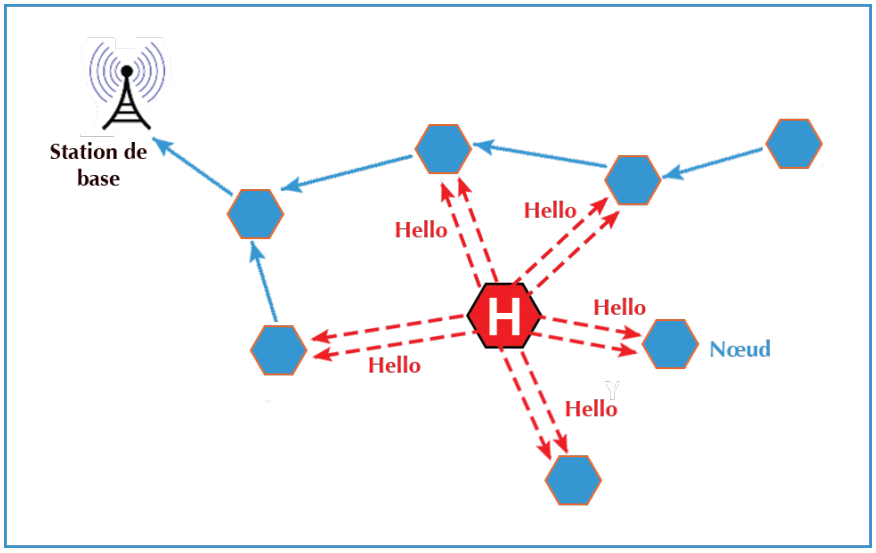

Figure 2. 6 : attaque Hello Flood

2.5.4.1 Les solutions de défense

Pour se défendre contre cette attaque, il est possible d'utiliser le mécanisme d'authentification par un capteur tiers. Plusieurs protocoles de routage nécessitent que les nœuds envoient des paquets, appelés paquets Hello, pour s'annoncer eux-mêmes à leurs voisins lors du déploiement. Un nœud qui reçoit un tel paquet va supposer qu'il est dans la portée du signal de l'expéditeur, et par la suite déduira que l'expéditeur est son voisin. Cette supposition peut être fausse : Dans la figure ci-dessus un adversaire possédant un haut potentiel de ressources (énergie et signal de transmission) est capable d'envoyer des paquets Hello à tous les capteurs afin de les convaincre qu'il est un nœud voisin avec le plus court chemin vers la station de base. Quand les capteurs vont envoyer des paquets vers la station de base, ils vont l'envoyer à celui de l'adversaire.

Cette attaque peut être aussi détectée par le calcul de RSSI. En télécommunications, le Received Signal Strength Indication où RSSI est une mesure de la puissance en réception d'un signal reçu d'une antenne (classiquement un signal radio). Son utilité est de fournir une indication sur l'intensité du signal reçu. [42].

2.6 L'attaque de retransmission sélective

L'attaque de retransmission sélective est une menace sérieuse dans les réseaux de capteurs sans fil, en particulier dans les systèmes de surveillance.

Les nœuds peuvent supprimer de manière malicieuse certain paquets de données sensibles, ce qui risque de détruire la valeur des données assemblées dans le réseau et de diminuer la disponibilité des services des capteurs.

Dans l'attaque de retransmission sélective, les nœuds malicieux refusent de transmettre certains messages provenant de leurs voisins en s'assurant qu'ils ne seront pas reproduits plus loin. Une forme simple de cette attaque interne peut être représentée par l'exemple suivant : un nœud malicieux se comporte comme un trou noir en refusant de transmettre tous les paquets qu'il reçoit. Toutefois, un tel attaquant prend le risque que les nœuds voisins le perçoivent comme ayant échoué à retransmettre les paquets et décident de chercher un autre chemin. Une forme plus subtile de cette attaque est lorsqu'un adversaire transmet les paquets d'une manière

sélective. C'est-à-dire qu'un adversaire intéressé à supprimer ou modifier les paquets provenant d'un nœud spécifique laisse passer d'une manière fiable le reste du trafic, ce qui limite les soupçons de ses méfaits. Les attaques de retransmission sélectives sont généralement plus efficaces lorsque l'attaquant est explicitement inclus sur le chemin d'un flux de données. Toutefois, il est concevable qu'un adversaire écoutant le flux passant par des nœuds voisins puisse être capable d'émuler la retransmission sélective en causant une collision sur chaque paquet transmis. Un adversaire lançant une attaque de retransmission sélective va probablement suivre le même chemin que le flux de données. Il peut donc efficacement s'installer sur le chemin du flux de données ciblées en se servant d'autres types d'attaques telles que Sinkhole.

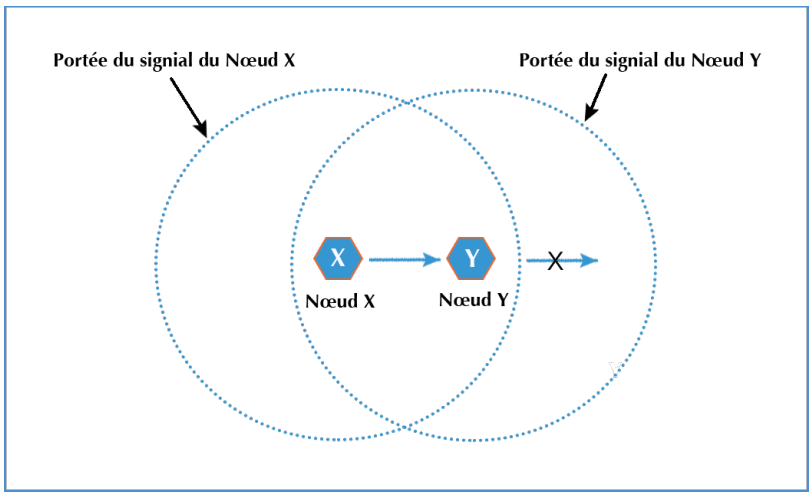

Figure 2. 7 : Attaque retransmission sélective

On prend cet exemple, en chemin vers la destination (la station de base), le nœud X envoie ou transmet un paquet à son nœud voisin Y.

L'attaque de retransmission sélective peut s'appliquer à partir du nœud malicieux Y de plusieurs façons :

- Scénario 1 : Y détruit le message.
- Scénario 2 : Y enregistre le message dans son cache et l'envoie plus tard causant un délai de transmission.
- Scénario 3 : Y retransmet le paquet à un autre nœud malicieux Z grâce à une connexion directe de haute qualité.

Dans les trois scénarios, le résultat peut être une perte du message sans détection ou un retard dans la transmission de données. Nous rappelons que dans une attaque de retransmission sélective, une portion des messages est ciblée et non pas tous les messages, ce qui rend la détection de cette attaque difficile. [33]

2.7 Attaque d'analyse de trafic

Dans le domaine du renseignement et de la sécurité informatique, l'attaque par analyse du trafic désigne les méthodes permettant de tirer des informations de flux de communication de tout type (émissions radio, trafic internet, mais également courrier papier, rencontres entre agents...) sans nécessairement avoir accès au contenu des messages échangés (notamment lorsque les messages sont cryptés). Ce type d'attaque peut être considéré comme une attaque par canal auxiliaire.

L'attaque d'analyse de trafic est une attaque passive, qui reconnait le patron (pattern) du trafic du réseau de capteurs sans fil, en analysant les mouvements des paquets dans le réseau. Ensuite, cette attaque déduit la localisation des capteurs stratégiques, et effectue une attaque de déni de service. La défense contre l'attaque d'analyse de trafic réside dans la prévention contre cette découverte de localisation. Les traditionnelles méthodes d'encryptage des données ne sont pas efficaces contre l'attaque d'analyse de trafic, car ces méthodes permettent seulement de cacher le contenu des données, et non la localisation de la station de base.

L'attaquant en interceptant des éléments sur la communication entre ses cibles peut déduire des indices sur leur activité.

Plusieurs données peuvent être étudiées par exemple :

Fréquence et volume des communications : Un trafic important peut indiquer un regain d'activité, la préparation d'une action. Inversement une baisse du trafic peut indiquer une diminution des ressources, une mise en sommeil de certaines cellules.

Distribution des messages : Une analyse expéditeur-destinataire peut donner des indications sur l'organisation de l'adversaire, par exemple une distribution descendante des messages ou l'existence de nouds correspondant avec de nombreux destinataires peut permettre d'identifier une chaîne de commandement, des va-et-vient rapides de messages peuvent indiquer une négociation.

L'évolution des indicateurs : Elle peut donner des indications, par exemple une diminution brutale des communications peut indiquer que l'adversaire soupçonne être surveillé et a choisi de limiter ses communications ou a choisi de basculer celle-ci sur d'autres modes.

Les menaces contre les réseaux de capteurs RCSF, comme l'attaque d'analyse de trafic, devient un élément important à prendre en compte pour le bon fonctionnement d'un réseau de capteurs sans fil. L'attaquant peut déduire les stations de base du réseau en observant les volumes de trafic et leurs formes (patterns). Ainsi, il peut effectuer une attaque de déni de service contre les capteurs stratégiques du réseau, créant une paralysie dans le réseau. La manière optimale de défense de la station de base contre l'attaque d'analyse de trafic est la modification de l'allure générale du trafic dans le réseau, en créant de nouvelles régions ayant un trafic volumineux.

La figure 2.18 montre le trafic au niveau de chaque capteur en utilisant le schéma SP (short path), qui détermine le chemin le plus court pour l'envoi des données à partir d'un capteur vers la station de base. Les capteurs, voisins de la station de base, renvoient plus de trafic que les autres capteurs.

En visualisant ce trafic, un attaquant peut déduire la région de la station de base. Par exemple:

- Si le contenu du message est un "texte simple" qui ne contient que des données, l'adversaire peut déterminer les paquets qui sont envoyés vers la station de base. Ce qui permet à l'attaquant de suivre la direction de ces paquets pour trouver la station de base.

- S'il existe une corrélation temporelle entre la réception d'un paquet par un capteur, et son renvoi, l'attaquant peut identifier ce paquet et le suivre saut par saut, jusqu'à la station de base.

- Comme la communication est élevée au voisinage de la station de base, un adversaire peut localiser cette dernière en suivant les régions de trafic élevé.

En général, un adversaire peut effectuer cette attaque de trois manières :

L'attaque par observation de taux (Rate Monitoring Attack) est basée sur le fait que les capteurs voisins de la station de base, envoient plus de messages que les capteurs éloignés de la station de base. Ainsi, un adversaire comptabilise le taux d'envoi des paquets, afin de connaitre les capteurs qui envoient le plus de paquets.

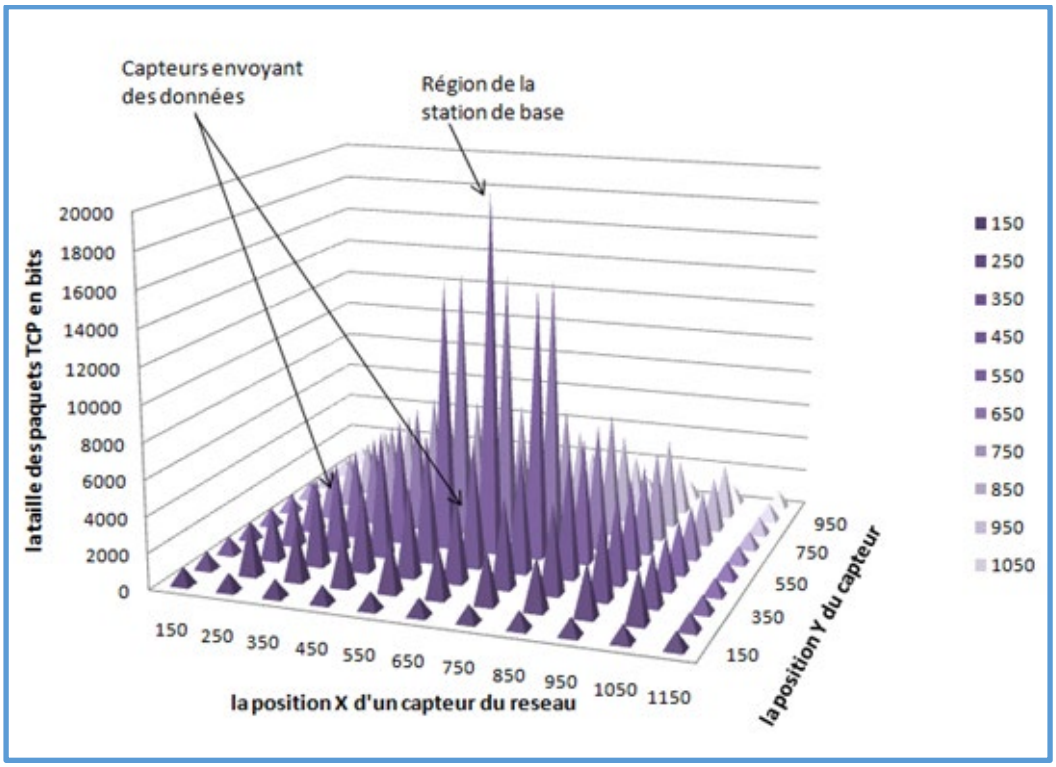

Figure 2.8 : Attaque analyse de trafic [43]

L'attaque par corrélation temporelle (Time Corrélation Attack) (appelée aussi l'attaque de traçage des paquets (packet tracing attack)) où l'attaquant calcule le temps d'envoi des paquets corrélés au travers des capteurs voisins, et essaye de tracer l'émission des paquets jusqu'à la station de base, ainsi l'attaquant suit un paquet à chaque saut jusqu'à la station de base.

L'attaque d'analyse des identifiants (ID) où un adversaire tente de détecter les relations entre les communications des capteurs, et ensuite en déduire le patron (pattern) du trafic en vérifiant les identités des paquets. [44]

2.7.1 Défense contre l'attaque d'analyse de trafic

Il y a plusieurs approches de défense contre l'attaque d'analyse de trafic, mais ces méthodes ne permettent pas une défense optimale, car elles consomment beaucoup d'énergie. Dans ce sens, les chercheurs se concentrent sur les techniques de dissimulation de la localisation de la station de base contre l'attaque du taux de trafic. A cette fin, ils utilisent trois techniques :

- Les routes multi-sauts des paquets en transmission, sont choisies de façon aléatoire.
- Certaines routes munies de faux paquets sont ajoutées dans le réseau.
- Plusieurs régions ayant un important faux trafic sont créées dans le réseau.

Cependant, ces techniques demeurent inefficaces contre l'attaque de corrélation temporelle. Jing Deng et al. [43] évaluent des contres mesures pour cacher la localisation de la station de base contre l'attaque d'analyse de trafic tels que :

- Le ré-encryptage des paquets à chaque saut pour changer leur apparence.
- La désignation d'un taux d'envoi de paquets uniforme.
- La suppression de la corrélation entre le temps de réception des paquets, et le temps de renvoi.

Jing Deng et al. [43] créent de multiples régions appelées (hot spots), dont ils démontrent l'efficacité analytiquement et par simulation en utilisant trois critères de mesure d'évaluation :

- L'entropie totale du réseau.
- L'énergie totale consommée du réseau.
- Le comportement de l'attaquant face aux contres mesures.

Par ailleurs, Ying et al. [45] proposent une défense contre l'attaque de corrélation temporelle appelée aussi attaque de traçage de paquets, en créant un trafic de données uniformément distribué dans tout le réseau de capteurs sans fil.

Afin de protéger la station de base, Xi Luo et al. [46] présentent une technique contre les trois types d'attaque d'analyse de trafic. Pour contrer l'attaque de surveillance du taux, ils choisissent aléatoirement un des capteurs voisins pour renvoyer les messages qui sont plus proches du capteur récepteur, afin de diminuer le temps de latence de transmission. Ils introduisent des faux paquets (dummy packet) qui sont ajoutés au trafic pour créer de la diversion, et enfin, ils utilisent des mécanismes d'anonymat pour cacher l'identité des capteurs qui participent à la transmission des paquets. Ils utilisent un réseau d'une centaine de capteurs, et considèrent un attaquant de type (localization evesdropper) qui détecte les transmissions par triangulation, et réside auprès d'une région afin de compter les paquets en transition. Ils proposent les deux techniques suivantes :

- **La technique RRS Random Routing Scheme** qui permet aux capteurs d'envoyer des paquets dans des directions différentes. Afin de calculer la probabilité d'envoi d'un paquet vers un capteur.
- **La technique DPIS Dummy packet injection Scheme** pour défendre une station de base contre l'attaque de surveillance par taux de paquets (Packet Rate Monitoring Attack), couplée à l'attaque de traçage de paquets (Packet Tracing Attack). Avec la technique DPIS, les faux paquets sont injectés avec une certaine probabilité proportionnelle à l'énergie résiduelle des capteurs.
- Ying et al. [45] proposent une technique LPR (**Location Privacy Routing Protocol**) qui défend la station de base seulement contre l'attaque de traçage de paquets. Ils combinent la diversification des routes avec l'injection de faux paquets, en utilisant un paramètre de probabilité de génération de faux paquets par un capteur réacheminant un vrai paquet.

2.8 Conclusion

Dans ce chapitre nous avons rappelé quelques types d'attaques connues qui visent un des objectifs de la sécurité comme l'authentification c'est le cas pour l'attaque Hello Flood. Dans ce sens nous avons cité les méthodes et les mécanismes utilisés pour se défendre contre ces attaques. Nous avons décrit l'attaque d'analyse de trafic et les défenses possibles contre cette attaque, mettant en place de nouvelles routes et générant du faux trafic.

A cause des caractéristiques limites des capteurs comme la faiblesse d'énergie, la dynamité de la topologie du réseau, la sécurité devient une tâche très difficile à réaliser parfaitement. Mais on peut diminuer le degré de danger de différentes attaques précisément qui sont de type

déni de service DoS en appliquant des solutions résistantes.

Dans ce contexte, on a introduit les réseaux de capteurs sans fil, leur utilité, leur mode de fonctionnement, ainsi que les différents concepts existants dans le domaine de sécurité d'un réseau sans fil. Ensuite, nous avons cité des solutions de défense capables de détecter quelques attaques dangereuses contre les réseaux de capteurs sans fil en se concentrant sur leurs principes de fonctionnement.

Finalement, on peut assurer que les réseaux de capteurs sans fil constituent un axe de recherche très fertile et peuvent être appliqués dans des domaines distincts. Cependant, il reste encore des problèmes à résoudre afin de se défendre contre les attaques soit actives ou internes en prenant en compte les aspects majeurs de la sécurité et de la protection fiable.

Chapitre

3 Analyse des performances pour le protocole d'agrégation de données sécurisées dans les réseaux de capteurs sans fil.

3.1 Introduction :

L'utilisation maximale de ressources limitées (énergie, bande passante et mémoire) dans les réseaux de capteurs sans fil constitue un défi pour la communauté des chercheurs. De plus, la sécurité dans la transmission de données est également une contrainte importante qui rend la recherche dans les RCSFs de plus en plus attrayante. L'agrégation de données est l'une des techniques de conservation de l'énergie qui minimise la surcharge informatique en éliminant les données redondantes. Cependant, certaines attaques de sécurité rendent les lectures d'agrégation de données fausses et ne fournissent donc pas des résultats précis. De plus, il existe un compromis entre sécurité et consommation énergétique de RCSF. Si nous optons pour une sécurité accrue, la consommation d'énergie augmente également (plus le cryptage et le décryptage entraînent une plus grande consommation d'énergie) et si nous essayons de préserver l'énergie, nous devons compromettre la sécurité quelque part.

Dans cet article, nous examinerons les protocoles d'agrégation sécurisée de données proposés, à la pointe de la technologie. Ces protocoles peuvent être classés et analysés lors de l'agrégation de données cryptées de bout en bout, et l'agrégation sécurisée de données non cryptées, sur la base du mécanisme de mise en œuvre de l'agrégation sécurisée de données. De plus, nous comparons les différents protocoles sur la base de performances telles que la consommation de communication et l'intégrité des données, etc. Enfin, nous avons discuté de certaines questions à étudier à l'avenir.

3.2 Agrégation dans les RCSFs

L'agrégation des données peut s'appliquer à une gamme d'opérations différentes à l'intérieur d'un réseau de capteurs sans fil. Elle est proposée comme un paradigme essentiel pour le routage sans fil dans les RCSF. Dans ce document, une définition du terme agrégation est proposée :

« L'agrégation des données est la tâche de rassembler des messages pendant qu'ils parcourent le réseau de capteurs en éliminant la redondance, en réduisant au minimum le nombre de transmissions et en économisant l'énergie dépensée ».

On distingue deux classes d'approches de base pour éliminer la redondance des données collectées. La première approche opère au niveau paquet, elle ignore les paquets qui ont été déjà circulé dans le réseau, ceci en associant deux paquets ou plus afin de réduire la charge des paquets, ou en utilisant des techniques de compression **(***ne rentre pas dans les objectifs de notre thèse***)**.

La deuxième approche opère au niveau application (le principe du *In-network processing*) et utilise la plate-forme du réseau pour faire des calculs sur les données (Max, Min, Avg, etc.).

Selon la deuxième approche, les données sont d'abord traitées dans le nœud de capteur avant qu'elles ne soient extraites pour une analyse externe. La combinaison des deux approches est possible.

L'agrégation est un concept essentiel qui permet la constitution d'une présentation synthétique des informations.

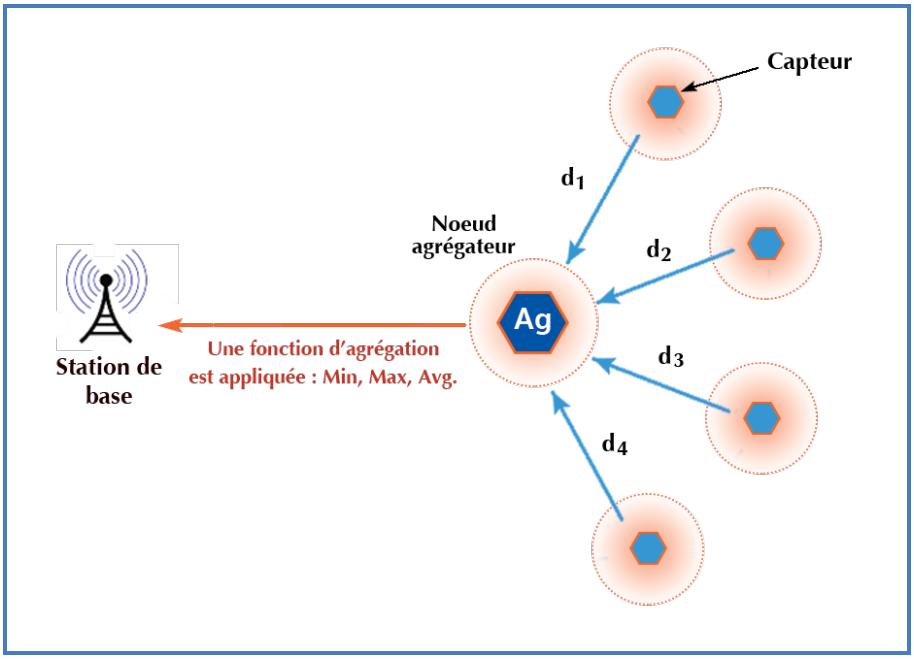

Figure 3.1 : Processus d'agrégation de données dans RCSF : *In-network processing*

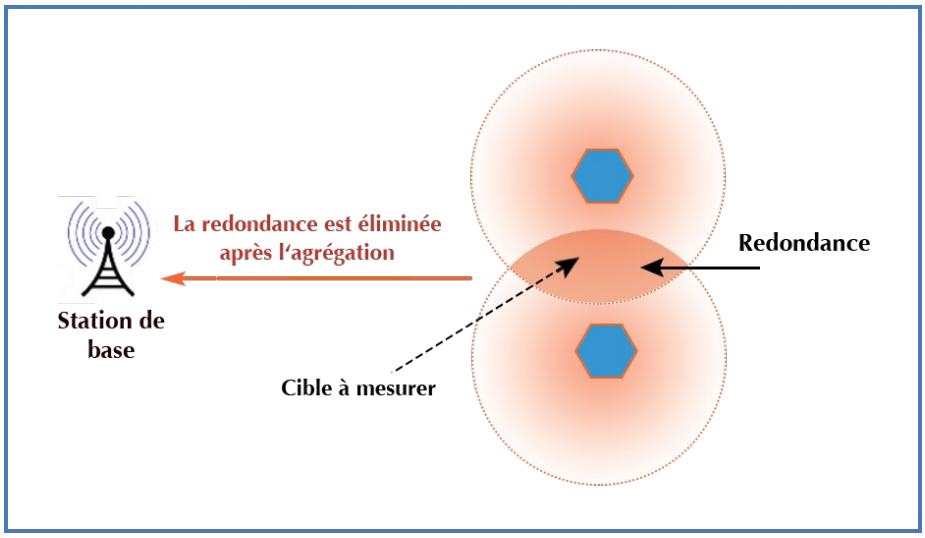

Figure 3. 2 : Processus d'agrégation de données dans RCSF : *Elimination de la redondance*

L'agrégation est considérée comme une fonction complémentaire d'un protocole de routage. De plus, elle dépend fortement de la nature des données collectées (corrélation, redondance) ainsi que du type d'application visé par le réseau.

Le but des différentes approches d'agrégation existantes est de maximiser des objectifs difficilement réalisables en même temps, i.e. on parle d'efficacité énergétique tout en assurant une qualité de service satisfaisante.

La fonction d'agrégation pourrait être exprimée, par exemple, par des requêtes SQL appropriées [47] :

```
SELECT {agg(expr), attributes} FROM sensors
WHERE {selection                icates}
GROUP BY {attributes}
HAVING {havingPredicates}
EPOCH DURATION i
```

Il existe de nombreuses catégories de fonctions d'agrégation parmi lesquelles nous pourrions énumérer [48] :

- ***Sensible aux doublons :*** Ces fonctions sont sensibles à la duplication. En effet, le résultat de la fonction est altéré si la valeur mesurée par un nœud est prise en compte plusieurs fois dans le calcul de la fonction. Cette catégorie comprend, par exemple, les fonctions SUM, MEDIAN et AVERAGE.
- ***Résumé :*** Une fonction de type résumé si son résultat dépend strictement de l'ensemble des valeurs enregistrées par les nœuds. Par exemple, la fonction SUM.
- ***Exemplaire :*** MIN et MAX sont dans la catégorie de fonction exemplaire.

Dans un processus d'agrégation de données par grappes (Fig. 3. 21), les nœuds sont d'abord regroupés en clusters gérés chacun par une tête de cluster. Ainsi, la fonction d'agrégation pourrait alors être calculée soit par une ou plusieurs têtes de grappes, soit, à défaut, par un ou plusieurs nœuds réguliers appelés agrégateurs.

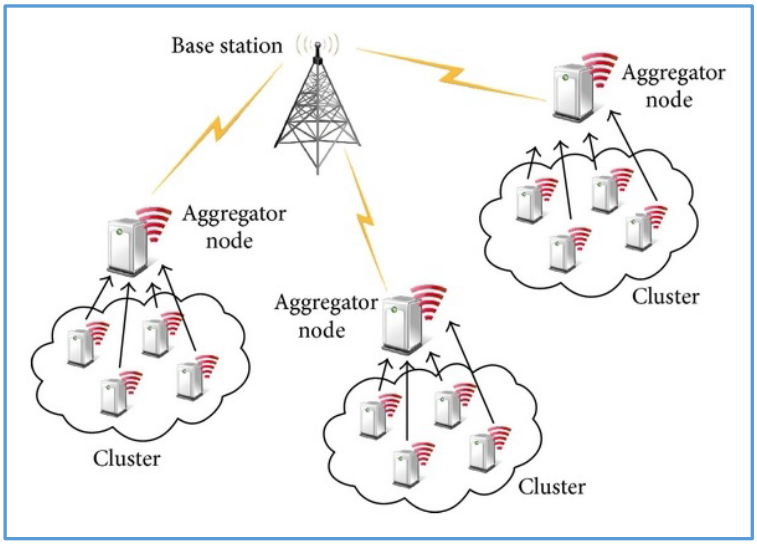

Figure 3.3 : Processus d'agrégation de données par grappes

3.3 L'agrégation et la sécurité des transmissions

La sécurité dans la transmission et l'agrégation de données est une question très importante à considérer lors de la conception d'un RCSF. Dans beaucoup d'applications, les nœuds capteurs sont déployés dans des zones ouvertes et sont exposés aux attaques qui pourraient espionner les informations échangées par les nœuds de capteurs.

3.3.1 Exigences de sécurité primaires

Atteindre la sécurité dans un réseau de capteurs sans fil est une tâche difficile en raison du déploiement hostile et de la nature limitée des ressources des nœuds de capteurs. Les problèmes

de sécurité liés à l'agrégation de données sont la confidentialité, la disponibilité, l'intégrité, l'authentification, la mise à jour des données, etc. Ce sont tous les besoins de sécurité du réseau de capteurs sans fil. De nombreux protocoles d'agrégation sécurisés ont tenté de répondre à ces exigences ensemble.

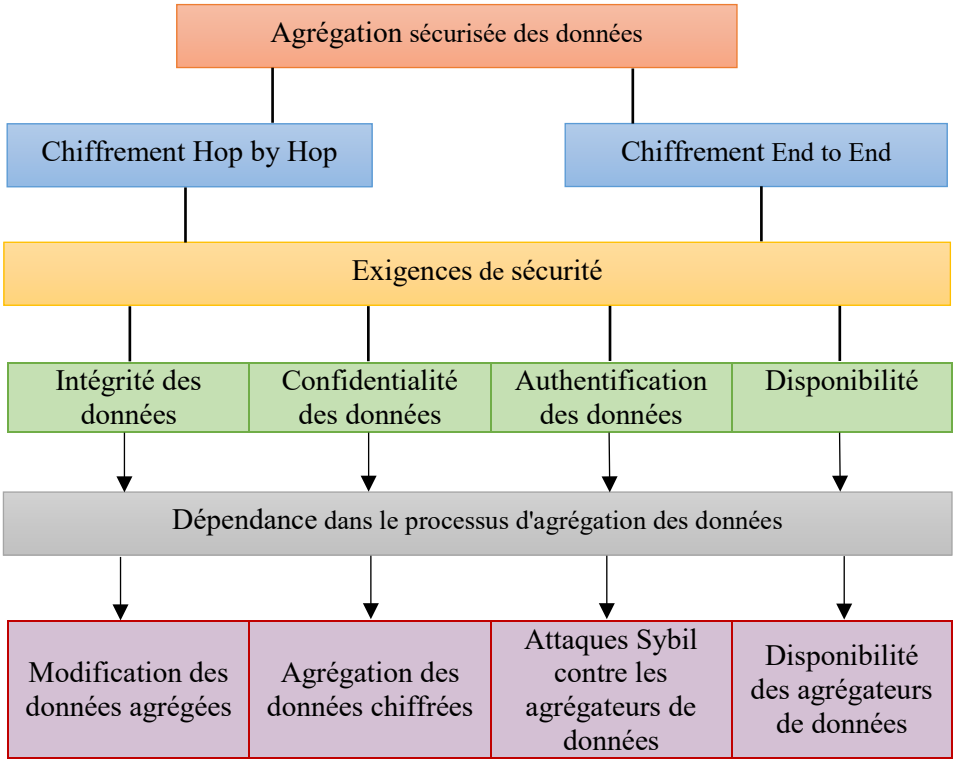

Tableau 3.1: Interaction entre la sécurité du réseau de capteurs sans fil et le processus d'agrégation de données.

3.3.2 Types d'agrégation sécurisée de données

L'agrégation de données saut par saut et l'agrégation de données de bout en bout sont deux méthodes utilisées pour l'agrégation de données sécurisées [50] dans les réseaux de capteurs sans fil.

Nous classons les protocoles d'agrégation proposés de données sécurisées en trois catégories : agrégation de données cryptées saut par saut, agrégation de données cryptées de bout en bout et agrégation de données sécurisées non cryptées.

- ***Agrégation de données cryptées saut par saut (hop-by-hop et de bout en bout (end-to-end)):***

La stratégie de confidentialité des données pour le processus d'agrégation de données dans RCSF peut être mise en œuvre de deux manières : saut par saut ou de bout en bout. Dans le premier cas, chaque nœud d'agrégation doit déchiffrer les données reçues, appliquer une fonction d'agrégation, chiffrer les données agrégées, puis les envoyer à un nœud d'agrégation supérieur en direction de la station de base. Cette méthode présente le principal inconvénient d'augmenter le temps de latence et les délais de transmission des données. Dans le modèle de bout en bout, les nœuds d'agrégation appliquent la fonction d'agrégation aux données cryptées reçues à l'aide de techniques de cryptage. Pour cette méthode, la latence, les retards de transmission puis la consommation d'énergie sont réduits.

3.3.3 Agrégation de données cryptées saut par saut :

- Le protocole SDAP (Secure Data Aggregation Protocol) est un protocole d'agrégation de données sécurisées saut par saut pour les réseaux de capteurs proposé par [51]. Ce schéma a un mécanisme pour représenter le réseau sous forme d'arborescence. Les auteurs ont choisi une technique probabiliste de partitionnement dynamique de l'arbre en sous-arbres, chacun d'entre eux contenant un nœud principal. La figure 3.4 montre un exemple d'arborescence d'agrégation SDAP. Les nœuds *x*, *y* et *w* avec la couleur grise foncé sont les nœuds principaux et la station de base en tant que racine est un leader par défaut [51]. Dans ce diagramme, chaque nœud feuille envoie ses données détectées à son parent, qui calcule une première opération d'agrégation de données avant d'envoyer les résultats d'agrégation au nœud principal de son groupe. Ce nœud principal applique ensuite une deuxième opération d'agrégation de données et envoie enfin le résumé de l'agrégation à la station de base. Le cryptage des données est effectué à l'aide de clés secrètes partagées entre chaque paire de nœuds, ce qui permet à SDAP de garantir l'intégrité et la confidentialité des données, ainsi que l'authentification du nœud source. Mais l'utilisation de l'énergie et les frais généraux de transmission sont élevées.

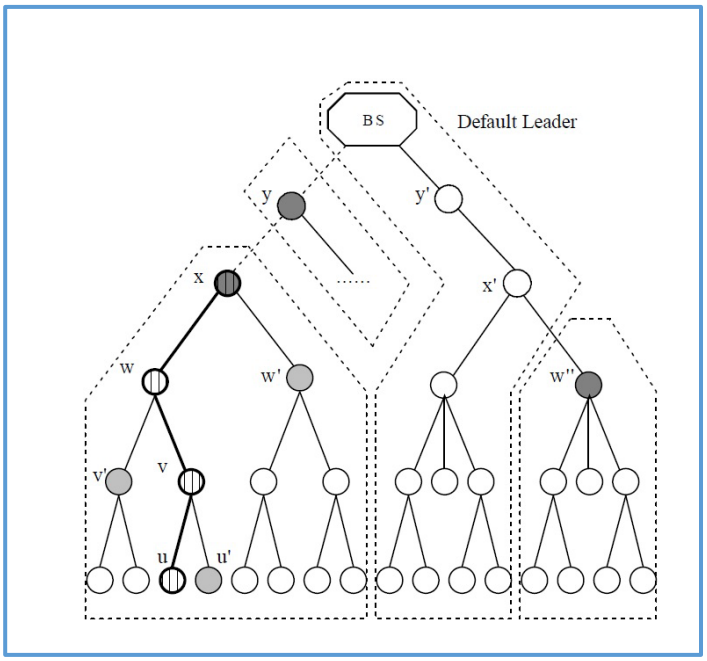

Figure 3.4 : Arbre d'agrégation SDAP

- Yoon et al [52] ont proposé un nouveau schéma d'agrégation de données sensibles (**NSDA** : New Sensitive Data Aggregation) pour protéger l'intégrité. Afin de préserver et de masquer les données détectées à partir de plusieurs nœuds, le protocole exploite des opérations arithmétiques au cours du processus d'agrégation et de transmission de données. Tous les nœuds utilisaient deux clés différentes, l'une était une clé secrète partagée avec un MD (dispositif principal) pour être un nœud de confiance du réseau, et l'autre était une clé symétrique partagée avec chaque nœud de capteur se trouvant dans leur arbre d'agrégation. Afin de réaliser des transmissions sécurisées.
- Kim Y. et al [53] ont proposé un protocole d'agrégation de données sécurisé (**HCDA**) basé sur la technique de la courbe de Hilbert et les échanges de graines entre les nœuds de capteurs. Premièrement, chaque nœud a déterminé ses nœuds frères, nœuds parents et nœuds enfants par schéma d'inondation. Afin de mettre en œuvre l'équilibre du réseau, HCDA a spécifié le nombre maximal de nœuds enfants pour un nœud. Ensuite, afin d'échanger des graines avec d'autres nœuds frères, chaque nœud a généré des données de graines aléatoires. La graine a été utilisée pour masquer les données détectées

d'origine. Les données détectées d'origine peuvent être modifiées en extrayant une partie d'une valeur de départ envoyée à d'autres nœuds. Une partie de la valeur de départ a également été ajoutée à partir d'un autre nœud. Enfin, les données détectées d'origine peuvent être masquées.

3.3.4 Agrégation de données chiffrées de bout en bout

- Le protocole CDAP est un schéma d'agrégation de données cachées utilisant l'homomorphisme de confidentialité proposé par [54]. Il est également basé sur un cryptage homomorphe pour assurer la sécurité des données agrégées. L'auteur déclare que le chiffrement homomorphe symétrique utilisé dans certains protocoles tels que [55] contient des problèmes de sécurité en raison de la clé partagée unique entre les nœuds. Pour cela, il utilise un chiffrement homomorphe asymétrique et en raison des coûts de calcul supplémentaire, le schéma utilise des nœuds puissants spéciaux appelés AGGNODE qui disposent de ressources suffisantes pour effectuer l'agrégation. Après le déploiement du réseau, chaque AGGNODE établit des paires de clés avec les nœuds autour de son voisinage qui peuvent ensuite envoyer leurs données détectées de manière sécurisée à l'AGNODE en suivant un algorithme de chiffrement symétrique. Lorsqu'un AGGNODE reçoit ses données d'agrégation, il les déchiffre, les agrège et chiffre le résultat avant de le transmettre à la station de base. Ce dernier peut ensuite décrypter le résultat de l'agrégation avec sa clé privée. Dans le protocole CDAP, la surcharge de calcul imposée par les fonctions de chiffrement homomorphique de confidentialité est tolérée en utilisant un ensemble de nœuds puissants (AGGNODE). Ainsi, le principal inconvénient de ce protocole est qu'il est particulièrement destiné aux réseaux de capteurs hétérogènes.

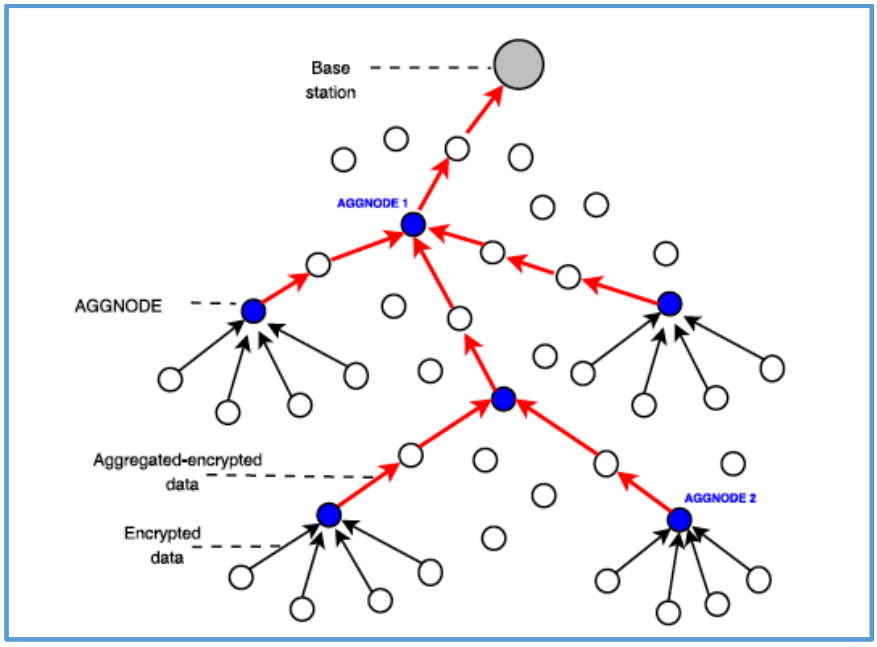

Figure 3.5 : Scénario d'agrégation du protocole CDAP. Les AGGNODE collectent des informations sur leur voisinage et les données chiffrées sont agrégées au niveau des AGGNODE pendant que les données se dirigent vers la station de base [54].

- Boubiche et al. [56] ont également proposé un protocole de regroupement de données sécurisé basé sur le tatouage numérique (**SDAW :** Secure data aggregation watermarking). Le filigrane a d'abord été intégré dans un espace fixe pour améliorer le

niveau de sécurité. Le paquet a été complété avec les données reçues pures. De cette façon, le filigrane serait difficilement intercepté par un attaquant.

Sur la base des recherches ci-dessus, lorsque les données détectées sont transmises d'un membre du cluster au nœud d'agrégation, la technique de cryptage homomorphe est la technologie la plus couramment utilisée pour garantir l'intégrité des données. Cependant, la génération et la distribution des clés, ces mécanismes impliquent des coûts de calcul supplémentaire et consomment plus d'énergie. Pour résoudre ce problème, le mécanisme a également utilisé une technique de filigrane fragile et légère sans chiffrement pour assurer l'authentification et l'intégrité des données détectées tout en économisant de l'énergie. Les liens entre les nœuds capteurs avec les nœuds d'agrégation et les liens entre les nœuds d'agrégation avec la station de base sont sécurisés à l'aide du mécanisme de tatouage (filigrane).

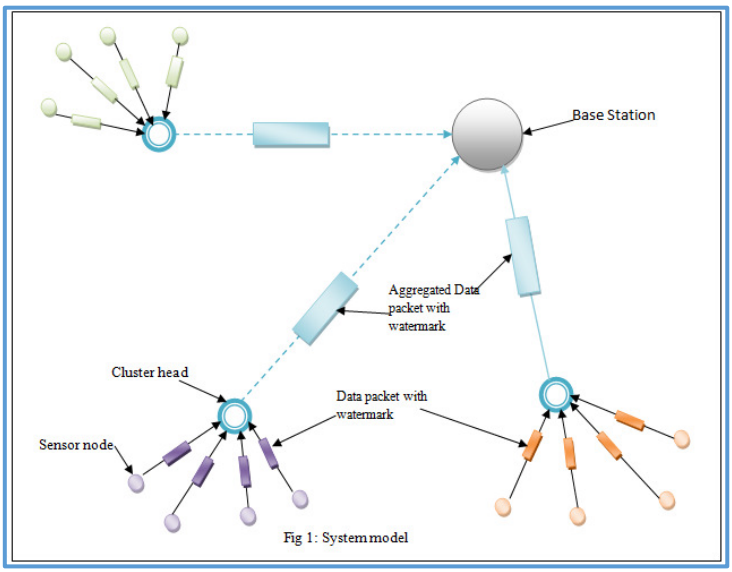

Figure 3.6 : model d'agrégation de données avec filigrane

- Omar et al. [57] ont proposé un protocole d'agrégation de données sécurisées pour RCSF basé sur un chiffrement homomorphe à clé publique (SASPKC: Secure Aggregation using Stateful Public Key Cryptography). Avant le déploiement du réseau, la BS génère sa paire clé public et privé. Chaque capteur était chargé d'une clé secrète partagée uniquement avec la BS, et également chargé d'un HMAC (keyed-Hash Message Authentication Code). Ensuite, chaque nœud a transmis son état à BS afin de générer les sous-clés nécessaires lors de la phase d'agrégation, et le CH (Cluster Head) a agi en tant que transitaire de données et non un agrégateur de données. Dans la phase d'agrégation des données, la valeur des données capturée par une clé a été codée avant le chiffrement, puis le texte chiffré a généré son MAC. Le CH a envoyé les données cryptées et les MAC au BS. Lorsque la BS a reçu tous les paquets de données, elle invoque les processus de déchiffrement et de vérification.

Les agrégateurs agrègent les lectures de capteur chiffrées sans les déchiffrer. Ainsi, le nœud d'agrégateur intermédiaire n'a pas besoin de vouloir stocker des informations secrètes, il offre donc une confidentialité de bout en bout entre les nœuds de capteur et le récepteur.

Paramètres	Hop-by-Hop Data Encryption	End-to-End Data Encryption
Confidentialité de bout en bout	Non	Oui
Delay (retard d'agrégation)	Agrégation avec retard	Agrégation sans retard
Intégrité des données	Fournit une intégrité maximale des données	Fournit une intégrité minimale des données
Agrégation effectuée	Données du capteur ordinaire	Données de capteur cryptées
Coût de calcul	Faible	Élevé
Mémoire nécessaire	Élevé	Faible
Vulnérable aux attaques	Plus à l'attaque passive	Plus à l'attaque active
Consommation d'énergie	Élevé	Faible
Sécurité des données	Moins sécurisée	Plus sécurisée

Tableau 3.2 : Comparaison des méthodes d'agrégation de données

3.3.5 Agrégation de données sécurisée non chiffrée

- Groat et al. [58] ont proposé un protocole d'agrégation de données sécurisées non chiffrées (KIPDA : k-Indistinguishable Privacy-preserving Data Aggregation), qui préservait la confidentialité des données agrégées en ajoutant un ensemble de valeurs de camouflage à l'ensemble de données détectées. Un ensemble de messages a été formé en combinant les données détectées par les nœuds de capteur et les données de camouflage. Le nœud du capteur a transmis les données détectées au nœud d'agrégation pour agréger les données par des fonctions non linéaires telles que MAX / MIN

- Protocole GPPS (Generic Privacy-Preservation Solutions) : Zhang W. et al [59] ont proposé un protocole d'agrégation de données sécurisé basé sur le modèle d'histogramme de perturbation. L'idée de base du protocole est de généraliser les valeurs des données transmises dans un RCSF, de telle sorte que bien que le contenu des données individuelles ne puisse pas être déchiffré, l'agrégateur peut toujours obtenir une estimation précise de l'histogramme de la distribution des données et ainsi rapprocher les agrégats. En particulier, avant la transmission, chaque nœud de capteur utilise d'abord une plage entière pour remplacer les données brutes. *MIN*, *MAX*, *médiane* et *histogramme*. Bien que le GP2S prenne en charge de nombreuses fonctions d'agrégation de données, elle présente les inconvénients suivants : Premièrement, le résultat agrégé final est une valeur d'approximation des données du capteur plutôt que des données réelles. Deuxièmement, le GPPS nécessite une charge utile de grande taille (message / données) car toutes les données du capteur doivent être remplacées par une plage entière. De plus, la consommation de bande passante de ce protocole augmente à mesure que le nombre de plages augmente. Enfin, le stockage des plages d'intervalles pour remplacer les données d'origine consomme une quantité importante de mémoire.

3.4 Technique de sécurité

Dans cette section, nous passons brièvement en revue certains mécanismes de sécurité, qui sont largement utilisés dans l'agrégation de données sécurisées dans les RCSFs.

3.4.1 Cryptage d'homomorphisme

Le cryptage homomorphe est un mécanisme de cryptage qui permet le calcul direct des données cryptées [60].

Soit Q et R désignent deux anneaux, + et × désignent respectivement additif et multiplicatif, et K représente l'espace clé.

E: $K \times Q \rightarrow R$ représente une opération de chiffrement. De même,

D: $K \times R \rightarrow Q$ représente une opération de décryptage. Si $a, b \in Q$, $k \in K$ et le cryptage d'homomorphisme additif et le cryptage d'homomorphisme multiplicatif sont présentés ci-dessous.

$$a + b = D_k (E_k(a) + E_k(b)) \quad (1)$$

$$a \times b = D_k (E_k(a) \times E_k(b)) \quad (2)$$

3.4.2 Signature numérique

Un des avantages majeurs de la cryptographie à clé publique réside dans la possibilité d'utiliser des *signatures numériques*. Une signature est une primitive cryptographique introduite par Diffie et Hellman et qui permet d'authentifier une donnée.

Les signatures numériques permettent à la personne qui reçoit une information de contrôler l'authenticité de son origine, et également de vérifier que l'information en question est intacte. Ainsi, les signatures numériques des systèmes à clé publique permettent *l'authentification* et le *contrôle d'intégrité* des données. Une signature numérique procure également la *non-répudiation*, ce qui signifie qu'elle empêche l'expéditeur de contester ultérieurement qu'il a bien émis cette information. (*On aura plus de détails sur la* signature numérique *dans le chapitre 4*).

3.4.3 MAC (code d'authentification de message)

MAC est une valeur obtenue par le résumé de clé et de message pour vérifier l'intégrité des données. Avant d'envoyer des données, l'expéditeur calcule le MAC des données via une fonction de hachage partagée avec le récepteur. Ensuite, les données avec MAC seront transmises au récepteur par l'expéditeur. Lorsque le récepteur a reçu le message, il peut récupérer les données et le MAC par clé de session et calcule le MAC des données reçues. Ensuite, le nouveau MAC et le MAC reçu seront comparés, s'ils sont égaux, le message est authentifié

3.4.4 Le tatouage numérique

Le tatouage numérique est une sorte de technologie dissimulée par les informations qui peut intégrer des informations spéciales dans les données originales [61]. Avant la transmission des données détectées, l'expéditeur calcule le tatouage numérique des données détectées via une disposition spéciale partagée avec le récepteur. Ces données sont transmises au récepteur. Le récepteur reçoit et exécute. Seules les données contenant le filigrane correct sont considérées comme fiables et seront traitées et transmises, sinon, elles seront rejetées.

3.4.5 Découpage de données

Dans l'arbre d'agrégation, les nœuds Leaf (nœuds feuille) sélectionnent de manière aléatoire un ensemble de nœuds dans la plage h hop. Dans les RCSFs à grande échelle, h peut être égal à 1. Lorsque les données C_i détectées par le nœud S_i, elles sont découpées en k blocs ($k = u$ désigne le nombre des nœuds du réseau) et les transmettent sur différents chemins. Les nœuds feuille transmettent non seulement des tranches de données, mais reçoivent également d'autres tranches de données. Les nœuds feuilles regroupent leurs propres données et les tranches de données reçues, puis les envoient au nœud supérieur.

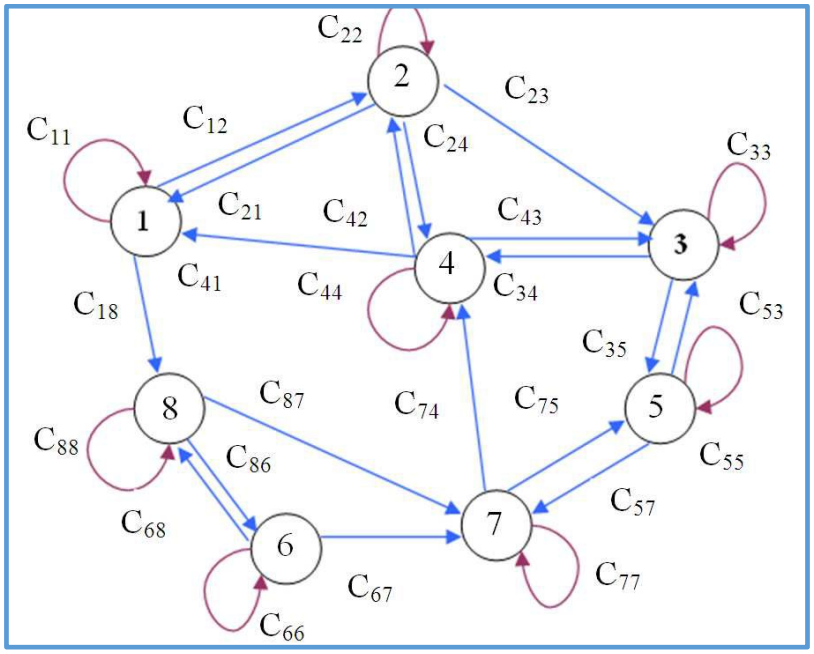

Figure 3.7 : Architecture de découpage (taille de réseau u = 8, longueur de saut $h_L = 1$)

Prenons le nœud 2 de la Figure 3.7. Lorsqu'il veut envoyer des données à ses nœuds voisins, il découpe les données (C) en 8 morceaux (car la taille du réseau u = 8). Les tranches restantes sont chiffrées avec leurs clés d'authentification respectives et les envoyés au reste des nœuds.

Lorsque le nœud 1 reçoit la tranche de données chiffrée à partir du nœud 2, il déchiffre la tranche à l'aide de sa clé d'authentification k_1. Ensuite, le nœud 1 attend la réception du reste des tranches jusqu'au temps t. Lorsque t expire, le nœud 1 cesse de recevoir la tranche de données. Après décryptage complet des tranches reçues, le nœud 1 les additionne avec la tranche qu'il contient et cette somme est représentée par S_1 : $S_1 = C_{11} + C_{21} + C_{41}$

De même :

$$S_2 = C_{22} + C_{12} + C_{42}$$
$$S_3 = C_{33} + C_{23} + C_{43} + C_{53}$$
$$S_4 = C_{44} + C_{24} + C_{74} + C_{34}$$
$$S_5 = C_{55} + C_{75} + C_{35}$$
$$S_6 = C_{66} + C_{86}$$
$$S_7 = C_{77} + C_{67} + C_{87} + C_{57}$$
$$S_8 = C_{88} + C_{68} + C_{18}$$

Le nœud 1 chiffre S_1 avec k_1 et envoyé à l'agrégateur A_1. L'agrégateur crypte les données avec la clé partagée secrète kj et les envoyées à la station de base.

3.5 Analyse de la sécurité et des performances

3.5.1 Analyse de sécurité

Cette section a comparé les différents protocoles d'agrégation de données sécurisées basés sur les technologies de sécurité et les fonctions d'agrégation, comme le montre le tableau 3.3.

Classification	Protocol	Security technologies	Fonctions d'agrégation
Hop-by-hop encrypted	SDAP	Chiffrement symétrique, MAC	MAX/MIN
	NSDA	Chiffrement symétrique, données masquées	SUM
	HCDA	Chiffrement symétrique, Courbe - Hilbert, l'échange de semences	SUM
End-to-end encrypted	CDAP	Chiffrement Homomorphe	Additive
	SDAW	Tatouage numérique (filigrane)	SUM
	SASPKC	Chiffrement à clé publique Stateful et le cryptage homomorphe	SUM
Non crypté	KIPDA	Perturbation des données	MIN/MAX
	GP^2S	Perturbation de données histogramme	MIN/MAX, SUM, AVE

Tableau 3.3 : Comparaison des protocoles d'agrégation de données

3.5.2 Analyse des performances

Nous évaluons les performances de chaque protocole en fonction des métriques suivantes :

- Coût de communication (CMC) : nombre de messages transmis dans l'ensemble des RCSFs.
- Coût de calcul (CPC) : il s'agit les surcoûts généraux du processeur pour obtenir une agrégation sécurisée des données.
- Précision des données (DA) : les données agrégées reçues au SB divisées par la somme des données originales réelles donnent le niveau de précision.
- Delay (DLY) : temps nécessaire pour obtenir les données détectées du nœud source vers la BS.
- Intégrité des données (DI) : il garantit que les données agrégées n'ont pas été modifiées par l'adversaire et d'autres nœuds malveillants lors de la transmission des données détectées du nœud source vers la station de base SB. il s'agit d'une mesure utilisée pour vérifier si un protocole prend en charge l'intégrité des données ou non. Si un protocole prend en charge cette fonction (la valeur DI est *Yes*), nous pouvons garantir que les données du capteur ont été correctement agrégées. Sinon, la valeur DI est *Non*.

Classification	Protocol	CMC	CPC	DA	DLY	DI
Hop-by-hop encrypted	SDAP	M	H	H	H	N
	NSDA	L	M	H	L	N
	HCDA	L	M	H	L	Y
End-to-end encrypted	CDAP	M	M	M	L	Y
	SDAW	L	M	H	M	Y
	SASPKC	L	H	H	L	Y
Unencrypted	KIPDA	L	L	H	L	Y
	GP^2S	L	M	L	L	N

Tableau 3.4 : Evaluations les performances des protocoles d'agrégation
Légende : L= Low, M = Medium, H = High, Y = Yes, N = No.

3.6 Conclusions

Dans les RCSFs, le protocole d'agrégation de données sécurisées réduit non seulement les données de trafic réseau et la consommation d'énergie, mais préserve également la sécurité des données dans une certaine mesure. Dans cet article, nous avons étudié de nombreux protocoles d'agrégation de données sécurisés différents. Ces protocoles peuvent être classés en agrégation de données chiffrées saut par saut, en agrégation de données chiffrées de bout en bout et en agrégation de données sécurisées non chiffrées. La technologie de sécurité et la fonction d'agrégation de différents protocoles sont analysées. De plus, les caractéristiques de performance de ces protocoles sont données sous la forme d'un tableau basé sur des métriques de performance. En principe, toutes les mesures susmentionnées sont importantes pour évaluer l'efficacité des protocoles mentionnés.

Nous soutenons que le coût de calcul (CPC), précision des données (DA), et l'intégrité des données (DI) sont les trois plus pertinents paramètres à considérer, pour avoir une agrégation de données privées. Remarquons que le protocole KIPDA répond à ces trois critères, car il a atteint une grande précision du résultat agrégé, bien qu'il ait besoin d'un coût de calcul relativement moins élevée, avec une garantie des données agrégées, mais les données ne sont pas chiffrées. Donc c'est très difficile d'avoir un protocole d'agrégation sécurisé parfait est efficace.

Enfin, nous discutons de certaines questions concernant l'agrégation sécurisée qui doivent être étudiées à l'avenir. Nous pensons que nos travaux aideront à concevoir des protocoles d'agrégation de données plus efficaces et sécurisées pour les chercheurs.

Chapitre

4 Un schéma de signature sans certificat amélioré basé sur RSA pour les réseaux de capteurs sans fil

4.1 Introduction :

La signature numérique est l'une des préoccupations les plus importantes dans les cryptosystèmes traditionnels à clé publique. Il existe plusieurs types de schémas de signature dans le développement de la technologie de signature : schémas de signature basés sur une infrastructure à clé publique (PKI), schémas de signature basés sur l'identité [76, 77] et schémas de signature sans certificat [78, 79]. Dans les premiers schémas de signature basés sur PKI, une autorité de certification (CA) de confiance est nécessaire pour générer un certificat qui correspond à la clé publique d'un utilisateur légitime. Par conséquent, ces systèmes doivent initialement obtenir et vérifier le certificat. Cela entraîne un certain nombre de coûts de calcul. Les schémas de signature basés sur l'identité excluent l'autorité de certification, mais la clé privée de chaque utilisateur légitime est produite et affectée secrètement par le générateur de clé privée (PKG), qui est entièrement approuvé. En conséquence, ce type de stratagème est vulnérable aux attaques lancées par le PKG. Les schémas de signature sans certificat (CLS) comblent les lacunes des deux types de schémas précédemment cités. Dans ce type de schéma, le centre de génération de clés (KGC) est chargé de calculer les clés privées partielles et d'attribuer les clés aux utilisateurs légitimes en secret. Chaque utilisateur calcule sa clé privée qui implique la clé privée partielle. Par conséquent, non seulement la légitimité de l'utilisateur peut être vérifiée, mais le KGC est également incapable de lancer une attaque en récupérant la clé privée de l'utilisateur. En cryptographie, les preuves des schémas de sécurité sont souvent liées à la complexité du calcul de certains problèmes mathématiques bien connus qui sont difficiles à résoudre. Donc, pour qu'un schéma fonctionne bien, il est nécessaire d'utiliser un modèle idéal, connu sous le nom de Random Oracle Model ou Black Box. C'est un modèle de calcul qui fournit des preuves de la sécurité d'un schéma cryptographique. Cet oracle aléatoire donne une réponse (True) à chaque requête unique à partir d'un domaine de sortie fixe au hasard.

Ainsi dans la contribution, que nous comptons présenter de ce chapitre, nous avons proposé un schéma de signature sans certificat basé sur RSA (Rivest Shamir Adleman), qui est un schéma largement appliqué dans des scénarios réels, en particulier pour les réseaux de capteurs sans fil. Nous avons montré que ce schéma est sécurisé dans le modèle d'oracles aléatoires et que sa sécurité est étroitement liée au problème de logarithme discret.

En effet, notre système offre une sécurité contre les attaques de type I, de type II. Il garantit également l'intégrité, la non répudiation et l'authentification.

L'organisation du reste de ce chapitre est la suivante : la section 1 contient les primitives cryptographiques utilisées dans les réseaux de capteurs sans fils. La section 2 couvre la signature numérique et la fonction de hachage. La section 3 présente le travail connexe et notre

contribution proposé basé sur RSA suivi d'une analyse de sécurité dans des oracles aléatoires, avec une analyse et de performance. La dernière section décline la conclusion et les travaux futures.

4.2 Primitives cryptographiques utilisées dans les réseaux de capteurs

Les mécanismes de sécurité du contrôle d'accès permettent de mettre en œuvre des services de sécurité. Dans la plupart des mécanismes de sécurité actuelle, la cryptographie est sans doute la technique la plus utilisée dans le cadre des réseaux filaires et des réseaux sans fil traditionnels, disposant d'une capacité de calcul et de mémoire conséquente. Les solutions de la cryptographie sont réputées comme des solutions sures qui répondent à l'ensemble des problèmes liés à la sécurité des données. Les spécificités des réseaux de capteurs, à savoir une faible puissance de calcul et une mémoire limitée auxquelles se rajoute la problématique de préservation de l'énergie, sont des freins considérables à l'utilisation des systèmes cryptographiques comme ECC, RSA, SSL, etc. Les travaux de recherche actuels s'attachent à trouver des solutions dites de cryptographie légères [62]. Ces solutions consistent à adapter les algorithmes de cryptographie classiques pour les réseaux de capteurs ou à trouver des nouveaux aussi efficaces en termes de sécurité, de temps d'exécution et de consommation énergétique. Deux types de cryptographie sont à distinguer : la cryptographie symétrique, appelée aussi à clé secrète, et la cryptographie asymétrique, ou à clé publique.

4.3 Le chiffrement symétrique et asymétrique

Il existe actuellement deux grands systèmes cryptographiques : ceux à chiffrement symétrique (dits à clé secrète ou privée) et ceux à chiffrement asymétrique (dits à clé publique).

4.3.1 Le chiffrement symétrique

D'une façon générale, l'expéditeur et le destinataire disposent de la même clé, l'un pour le déploiement de l'algorithme de chiffrement et l'autre pour le déroulement du processus de déchiffrement. Les deux algorithmes sont inverses l'un de l'autre.

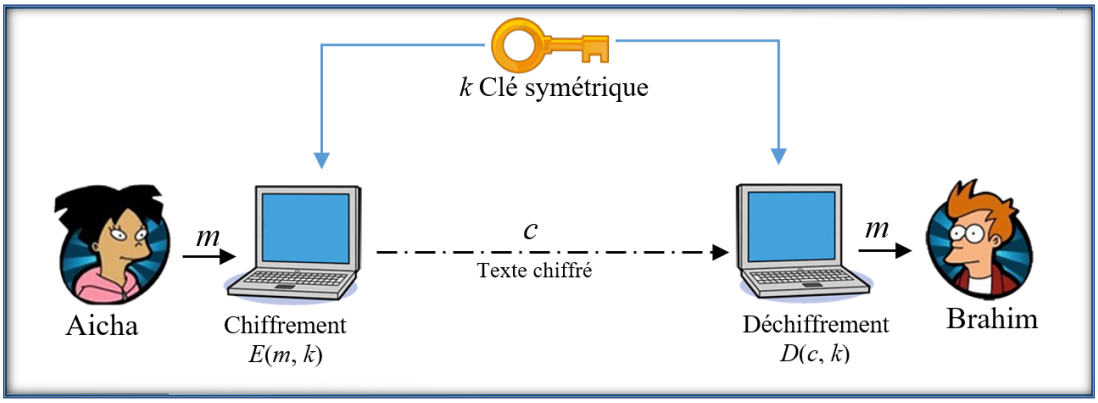

Figure 4.1. Schéma du Chiffrement symétrique

Dans les RCSF, le chiffrement considère que les deux nœuds impliqués dans la communication, par exemple N_1 et N_2, partagent une clé secrète, par exemple k. Cette clé permet aux deux nœuds à la fois de chiffrer et de déchiffrer les données :

$$\text{Chiffrement} : E : K \times M \longrightarrow C, \qquad c = E(k, m)$$

$$\text{Déchiffrement} : D : K \times C \longrightarrow M, \qquad m = D(k, c)$$

où K est l'ensemble de toutes les clés k possibles, M est l'ensemble de tous les messages

possibles, C est l'ensemble de tous les messages chiffrés possibles, m et c sont respectivement le message choisi et son chiffré correspondant.

L'algorithme symétrique le plus célèbre est le DES (Data Encryption Standard), qui fonctionnait avec des clés de 64 bits. Il a été remplacé par l'AES (Advanced Encryption System), qui fonctionne avec des clés allant jusqu'à 256 bits.

Les algorithmes symétriques sont très rapides en termes de calcul. Cependant, ils posent le problème de distribution de clés entre un émetteur et un récepteur. Le partage d'une clé avec chaque nœud communicant dans un groupe de N nœuds est difficile et conduit à un grand nombre de clés à gérer : $N(N–1)/2$.

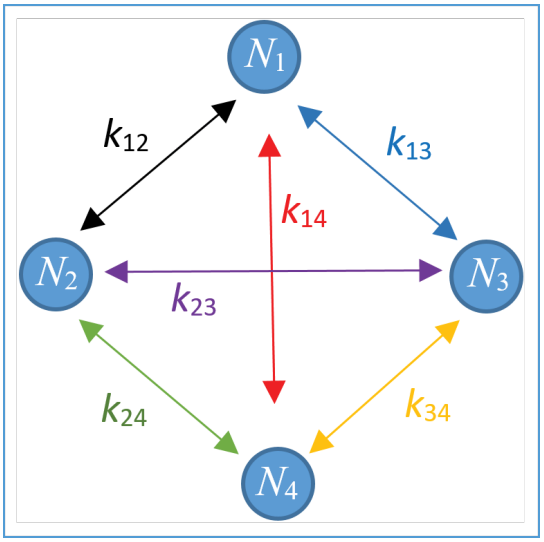

Figure 4.2 Exemple de gestion des clés si N=4

4.3.2 Le chiffrement asymétrique

La cryptographie asymétrique se base sur le principe de deux clés : clé publique et clé privée. La clé publique est mise à la disposition de quiconque désire chiffrer un message (cette clé peut être connue par tout le monde). Ce message chiffré ne pourra être déchiffré qu'avec la clé privée, qui doit être confidentielle et connue seulement par son propriétaire.

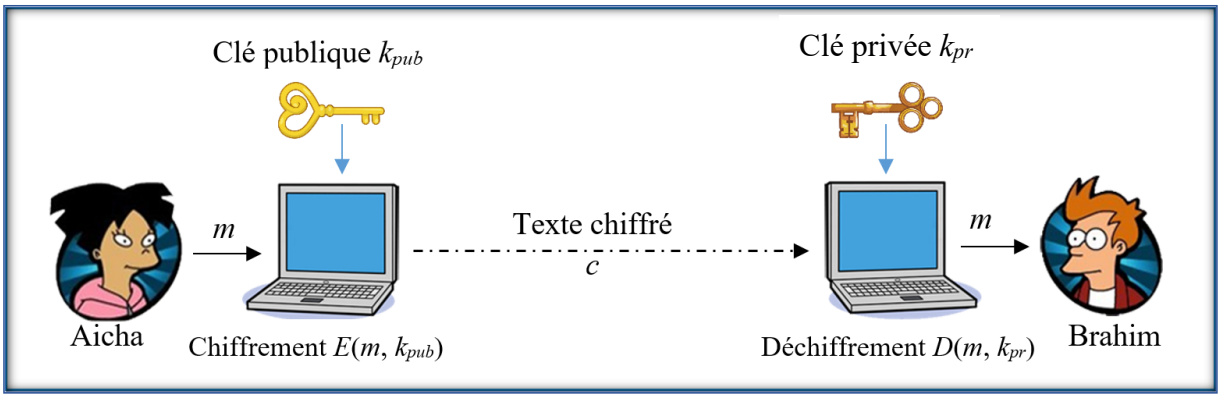

Figure 4.3 : Chiffrement asymétrique : Aicha utilise la clé publique de Brahim afin de lui envoyer un texte chiffré.

Par exemple dans la figure 4.3, Aicha possède une clé publique k_{pub} et Brahim une clé privée k_{pr}. Tout le monde peut envoyer des messages à Brahim en les chiffrant avec la clé publique, seule Brahim peut les déchiffrer avec sa clé privée.

Un chiffrement asymétrique est défini par trois algorithmes :

- Algorithme de génération des clés ;
- Algorithme de chiffrement ;
- Algorithme de déchiffrement.

Pour le cryptage de la meilleure méthode consiste à combiner les systèmes de clé privé et de clé publique de façon à avoir les avantages concernant la sécurité des clés publiques ainsi que la rapidité des clés privées.

Le tableau suivant présent une comparaison entre les deux systèmes :

Types de chiffrement	Avantages	Inconvénients	Complexité	Algorithmes employés
Symétrique	▪ Clé unique pour le chiffrement/déchiffrement ▪ Rapidité du chiffrement/déchiffrement ▪ Clés relativement courtes (128 ou 256 bits)	▪ Difficulté de gestion des clés (nombreux de pair de clés : $N.(N–1)/2$ clés) ▪ Certaines propriétés (p.ex. signatures) sont difficiles à réaliser	$O(\log(N))$ (AES)	RC4 AES DES 3DES QUAD
Asymétrique	▪ Une clé connue par tout le monde et une clé personnelle ▪ Aucun transfert de clé privée ▪ Quiconque peut envoyer un message chiffré unique pour le chiffrement/déchiffrement	▪ Authentification incertaine de l'expéditeur ▪ Algorithme lent (100 à 1000 fois que certains algorithme symétrique) ▪ Longueur des clés	$O(N^3)$ (RSA)	RSA Diffie Hellman ECC El Gamal DSA

Tableau 4.1 : Comparaison entre le chiffrement Symétrique et Asymétrique

L'un des principaux avantages de la cryptographie de clé publique est qu'elle offre une méthode d'utilisation des signatures numériques.

4.4. La signature numérique

La signature numérique est un mécanisme qui permet d'authentifier un message, autrement dit de prouver qu'un message provient bien d'un expéditeur donné.

Les signatures numériques permettent à la personne qui reçoit une information de contrôler l'authenticité de son origine et également de vérifier que l'information en question est intacte. Ainsi, les signatures numériques des systèmes à clé publique permettent l'authentification et le contrôle d'intégrité des données. Une signature numérique procure également la non-répudiation, ce qui signifie qu'elle empêche l'expéditeur de contester ultérieurement qu'il a bien émis cette information.

Une signature numérique a la même utilité qu'une signature manuscrite. Cependant, une signature manuscrite peut être facilement imitée, alors qu'une signature numérique est

pratiquement infalsifiable.

La signature d'un document utilise à la fois la cryptographie asymétrique et les fonctions de hachage.

4.4.1 Fonctions de Hachage

Une fonction de hachage *H(.)* est une fonction qui transforme un long message *m* en un résumé court, de taille fixe. Le message *H(m)* rendu par la fonction de hachage s'appelle *le condensé du message*, *l'empreinte du message*, *le résumé du message* ou encore *le message haché* [63][64]. Le point fort de la fonction de hachage est qu'elle peut être aisément calculée mais difficilement inversée. De plus, la moindre modification du message original entraîne un changement dans le condensé. Grâce à ses propriétés, le hachage a été employé dans les mécanismes de sécurité tels que la signature numérique.

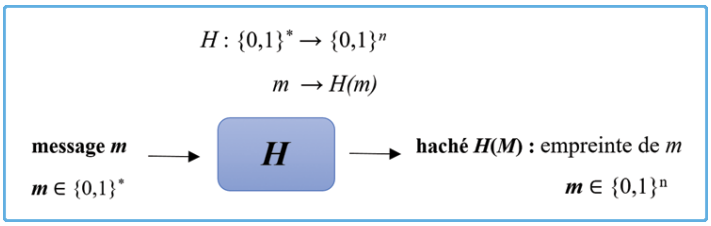

Figure 4.4 : Une fonction de hachage *H* calcule un haché de *n* bits à partir d'un message arbitraire *m*

- Propriétés des fonctions de hachage cryptographique :

- **Résistance à la collision :** Il est difficile de trouver un *m* et *m'* tels que *H(m) = H(m').*
- **Fonction de Hachage à sens unique :** Il est très difficile de retrouver ou générer un message *m* à partir de l'empreinte *H(m)* (on parle alors de fonction à sens unique).
- **Résistance à la préimage :** Etant donnée $y = H(m)$, il est difficile de trouver $x = m$ tel que $y = H(m)$ (Exemple : soit $H(m) = 2^n$, calcul de hash requis pour trouver *m*, c'est difficile).
- **Résistance à la seconde préimage :** Etant donné $x = m$, il est difficile de trouver $m' \neq m$ tel que $H(m) = H(m')$.

Selon sa définition, une fonction de hachage est une fonction dont l'ensemble de départ est plus grand que l'ensemble d'arrivée. Théoriquement, l'ensemble de départ peut être infini; en pratique, l'ensemble de départ comprend généralement tous les messages d'une taille inférieure à un certain seuil.

L'existence de collisions est alors inévitable pour une fonction de hachage, *H*, donnant des empreintes de taille *n.* Si on choisit $2^n + 1$ messages distincts, il existe forcement une paire de messages aboutissant au même haché.

- Paradoxe des anniversaires :

Ce paradoxe désigne un phénomène contre-intuitif : dans un ensemble de 23 personnes choisies aléatoirement, la probabilité que deux personnes fêtent leurs anniversaire le même jour de l'année est supérieure à 1/2. Ce comportement inattendu peut néanmoins être expliqué en suivant le raisonnement suivant :

Soit *k* éléments $x_1, x_2, \ldots, x_k$ tirés uniformément et indépendamment dans un ensemble *E* de taille *n*. La probabilité que tous les x_i soient distincts est :

$$\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\left(1-\frac{3}{n}\right)\ldots\left(1-\frac{k-1}{n}\right)=\frac{n!}{(n-k)!}\cdot\frac{1}{n^k}$$

Par conséquent, la probabilité qu'au moins deux éléments soient identiques est

$$p = 1 - \frac{n!}{(n-k)!} \cdot \frac{1}{n^k} \approx 1 - e^{-\frac{k(k-1)}{2n}}$$

De ce fait, pour le cas des anniversaires, on peut constater que pour $n = 365$ et $k = 23$, cette probabilité devient proche de 1/2, voir la figure 4.5.

En appliquant ce principe dans le cas des fonctions de hachage, pour trouver une collision dans un ensemble de taille 2^n, il faudra essayer $2^{n/2}$ valeurs distinctes pour produire une collision avec une probabilité supérieure à 1/2. Dans le pire des cas, un attaquant à besoin $2^{n/2}$ de calculs, aussi une mémoire de $2^{n/2}$ [65].

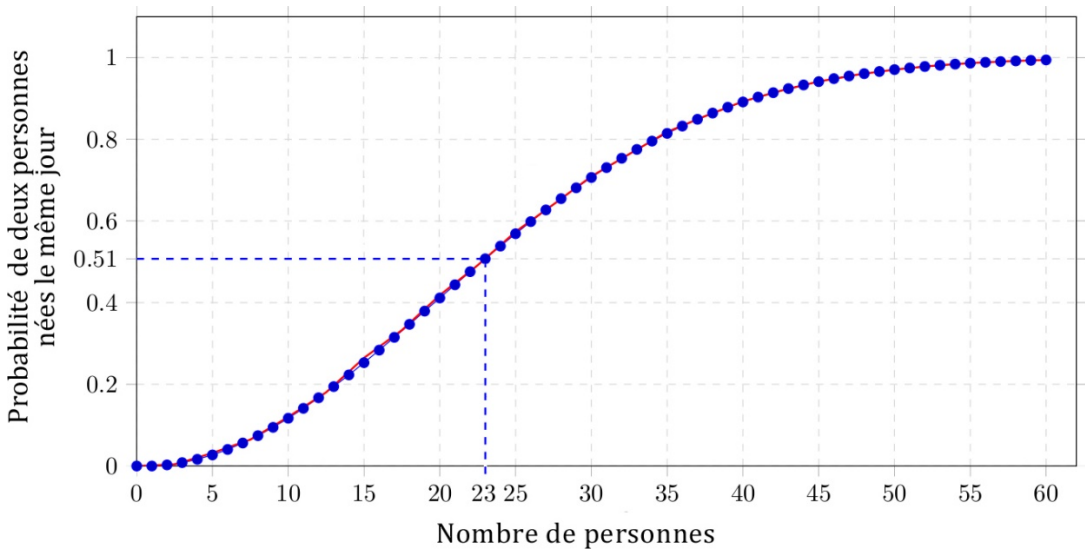

Figure 4.5 : La chance sur deux pour que deux personnes au moins soient nées le même jour de l'année

Les complexités des attaques génériques pour le cas des collisions, des préimages et des deuxièmes préimages sont rassemblées dans le tableau 3.2.

Attaque générique	Complexité
Recherche de préimages	$O(2^n)$
Recherche de deuxièmes préimages	$O(2^n)$
Recherche de collisions	$O(2^{n/2})$

Tableau 4.2 : Complexité des meilleures attaques génériques

Une des applications les plus importantes des fonctions de hachage est leur utilisation dans les signatures numériques.

4.4.2 Fonctionnement de la signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul

ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur, la figure 3.5 montre les étapes de la signature numérique.

Un bon protocole de signature électronique doit avoir les propriétés suivantes :

- Il doit garantir *l'authenticité* : la signature ne doit pas pouvoir être imitée. Autrement dit, si un document porte la signature électronique d′Aicha, on doit pouvoir être sûr que c'est effectivement Aicha qu'elle a signé.
- Il doit garantir *la non-répudiation* : si Aicha a signé un document, elle ne doit pas pouvoir se rétracter et pouvoir prétendre ensuite qu'elle ne l'a pas signé.
- Il doit garantir *l'intégrité* : en comparant la signature et le message, on doit pouvoir être sûr que le message n'a pas été altéré lors de l'envoi

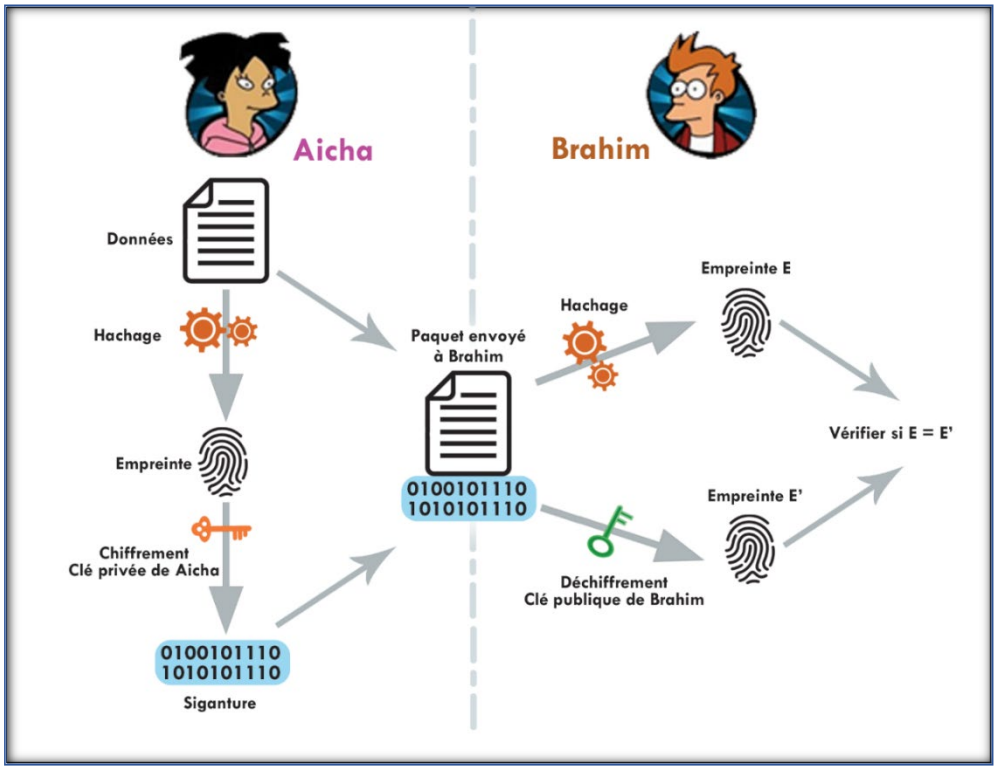

Figure 4.6 : Signature d′un message et sa vérification

4.4.3 Certificats numériques

Un problème, avec les crypto-systèmes à clés publiques, est que les utilisateurs doivent être constamment vigilants pour s'assurer qu'ils chiffrent leurs messages en utilisant la véritable clé de leur destinataire.

Un certificat est un document électronique, résultat d′un traitement fixant les relations qui existent entre une clef publique, son propriétaire (une personne, une application, un site) et l′application pour laquelle il est émis.

Version : la version de X.509 utilisé
Serial Number : numéro de séquence du certificat (propre à chaque autorité de certification)
Signature Algorithm : algorithme utilisé pour signer
Issuer : le nom du CA qui a émis le certificat
Validity : période de validité
Subject : le nom du propriétaire de ce certificat
Public Key : clé publique de l'utilisateur
Subject Public Key Info : des informations sur la clé publique (type, longueur…etc.)
Signature numérique : signature du CA sur l'ensemble des champs précédents.

Figure 4.6 : Contenu d'un certificat

4.4.3.1 Autorité de Certification :

Une Autorité de Certification (CA : Certification Authority) est une organisation qui délivre des certificats électroniques à une population. En délivrant un certificat, le CA se porte garant de l'identité de l'entité qui se présentera avec ce certificat. Par rapport aux entités (personnes ou applications) qui utilisent ses certificats, une CA joue le rôle de tiers de confiance.

Une CA utilise sa clé privée pour créer les certificats qu'elle délivre. Le certificat est signé (au sens signature électronique) : on effectue une empreinte (ou un condensé) du certificat à l'aide d'un algorithme de hachage et on chiffre l'empreinte obtenue. Le chiffrement s'effectue avec la clé privée de l'autorité de certification.

4.4.3.2 Vérification d'un certificat

La vérification s'effectue avec la clé publique de l'autorité de certification. Toute personne voulant vérifier un certificat présenté par une entité doit connaître la clé publique de l'autorité de certification. Elle procède ensuite de la même façon que la vérification d'une signature (déchiffrer la signature avec la clé publique du CA puis comparer le résultat avec le résumé des informations figurant dans le certificat).

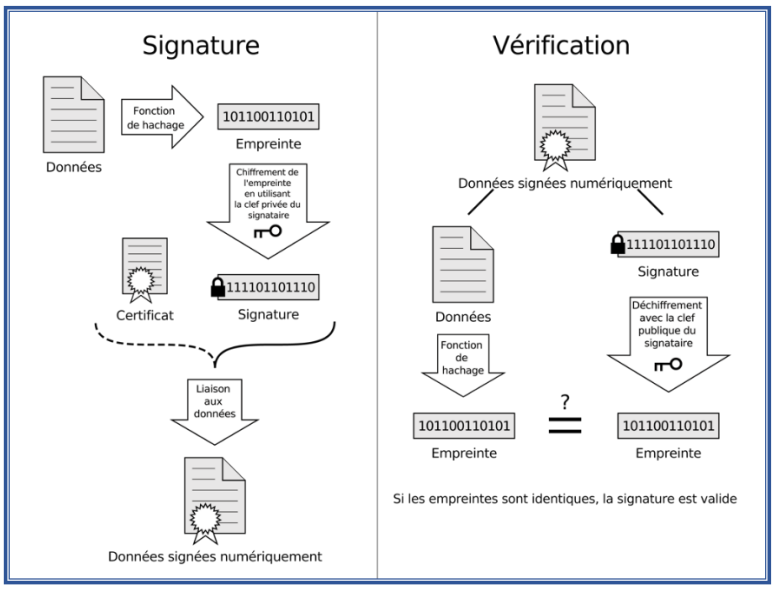

Figure 4.7 : Anatomie d'une signature numérique avec certificat

4.5 Le cryptosystème RSA

RSA (Rivest Shamir Adleman) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman [66].

La factorisation est un problème algorithmique dont la solution est difficile à trouver. Il est facile de trouver deux grands nombres premiers p et q et de calculer leur produit $n = p \times q$. En revanche, le problème consiste à retrouver p et q en connaissant n quand p et q sont plus grands. Ce problème est appelé la factorisation.

4.5.1 Génération des clefs :

Le RSA fonctionne à partir de deux nombres premiers, que l'on appellera p et q. Ces deux nombres doivent être très grands, car ils sont la clé de voûte de notre cryptage.

Une fois ces deux nombres déterminés, multiplions-les. On note n le produit $n = p.q$, et on calcule l'indicateur d'Euler $\varphi(n) = (p-1)(q-1)$.

Cherchons maintenant un nombre e (inférieur à $\varphi(n)$), qui doit nécessairement être premier avec $\varphi(n)$. Calculons ensuite l'inverse de e modulo $\varphi(n)$, que nous noterons d, avec $d \equiv e^{-1} \bmod \varphi(n)$ [67].

Alors :

La clé publique : c'est le couple (n, e).
La clé privée : c'est le couple (n, d).

Algorithme 4.1: Génération des clés RSA

Entrée: La taille l du module. // Taille en bit qui détermine le niveau de sécurité.
Sortie: Une clé publique (n, e) et une clé privée (n, d).
1. Générez deux grands nombres premiers p et q aléatoires et distincts $(l / 2)$-bits.
2. Calculez $n = pq$ et $\varphi(n) = (p-1)(q-1)$.
3. Choisissez un entier aléatoire e tel que $3 \leq e < \varphi(n)$ et pgcd $(e, \varphi(n)) \equiv 1$.
4. Calculez l'entier unique d tel que $1 \leq e < \varphi(n)$ et $ed \equiv 1 \pmod{\varphi(n)}$.
5. Retourner la clé publique (n, e) et la clé privée (n, d).

4.5.2 Les algorithmes de chiffrement, déchiffrement, signature et vérification de la signature :

Avec la clé publique (n, e), on calcule $c = m^e \bmod n$, où m est entre 0 et $n-1$ est le message à chiffrer et c le message chiffré transmis.

Algorithme 4.2: Chiffrement RSA

Entrée: La clé publique (n, e) et le texte en clair m. // La clé publique et le texte claire $m \in [0, n-1]$
Sortie: Le texte chiffré c.
1. Calculez $c \equiv m^e \bmod n$.
2. Retourner le texte chiffré c

Algorithme 4.3: Déchiffrement RSA

Entrée : La clé privée (n, d) et le texte chiffré c.

Sortie : Le message m.

1. Calculez $m \equiv c^d \bmod n$.
2. Retourner le message m.

Algorithme 4.4: Signature avec RSA

Entrée : La clé publique (n, e), la clé privée d et le message m.

Sortie : s. // *La signature*

1. Calculez $h = H(m)$.
2. Calculez $s = h^d \bmod n$.
3. Retourner (s).

Vérification de la signature : A la réception de la signature, l'autre partie calcule $h=H(m)$, puis à partir de la clé publique et de la signature s reçue, elle calcule $h' = s^e \bmod n$. Enfin, elle accepte la signature si $h=h'$.

Algorithme 4.5: Vérification de la signature

Entrée : La clé publique (n, e), m et s.

Sortie : *Acceptation ou rejet de la signature.*

1. Calculez $h = H(m)$.
2. Calculez $h' = s^e \bmod n$.
3. Accepter si $h=h'$ *et rejet si non.*

4.5.3 Exemple d'illustration du cryptosystème RSA

- **Génération des clés :**

  Soient $p = 3011$ et $q = 3037$, On trouve $n = p.q = 9144407$,

  Par la suit on :

  - Calculons l'indicatrice d'Euler $\varphi(n) = \varphi(p).\varphi(q) = (p-1).(q-1) = 9138360$.
  - Cherchons une valeur e inferieur à $\varphi(n)$, et qui doit être premier avec cette dernière. Par exemple on prend $e = 7185583$.
  - Calculons d (l'inverse de e modulo $\varphi(n)$), on trouve $d = 7449007$.

  On sélectionne :

  - La clé privée : c'est le couple (9144407, 7449007).
  - La clé publique : c'est le couple (9144407, 7185583).

- **Chiffrement :**

La relation de chiffrement est : $c = m^e \bmod n$.

Aicha désire envoyer un message m secret à Brahim. Ce dernier lui a fait connaître ses clefs publiques et elle va les utiliser pour chiffrer le texte suivant : m = « IL FAIT BEAU ».

Tout d'abord, il faut convertir le message à envoyer en chiffres, et pour cela on considère que chaque alphabet équivalent a deux nombres comme le présent le tableau suivant :

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
00	01	02	03	04	05	06	07	08	09	10	11	12
N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25
Espace												
26												

Tableau 4.3 : Code de conversion

Donc le message m devient m' = 081126 050008 192601 040020 .
Comme le message m doit être inférieur à n–1 = 9144406, il faut qu'on décompose le message m' en blocs inferieur à n – 1. Alors m' devient m_1=081126, m_2=050008 , m_3=192601 et m_4= 040020.

Et par la suite, Aicha chiffre les blocs obtenus par la relation $c = m^e \bmod n$ avec le couple (e, n) est la clé publique de Brahim, et on obtient les résultats suivants :

$$c_1 = (081126)^{7185583} \bmod (9144407) = 6016185.$$
$$c_2 = (050008)^{7185583} \bmod (9144407) = 4658242.$$
$$c_3 = (192601)^{7185583} \bmod (9144407) = 7819676.$$
$$c_4 = (040020)^{7185583} \bmod (9144407) = 7530290.$$

Aicha envoie à Brahim le message chiffré suivant : c = 6016185 4658242 7819676 7530290.

- **Déchiffrement**

La relation de chiffrement est : $m = c^d \bmod n$.
Brahim reçoit le message c, et utilise sa clé privé (d, n) pour déchiffrer le message c.
On trouve les résultats suivants :

$$m_1 = c^d \bmod n = 6016185^{7449007} \bmod 9144407 = 081126.$$
$$m_2 = c^d \bmod n = 4658242^{7449007} \bmod 9144407 = 190417.$$
$$m_3 = c^d \bmod n = 7819676^{7449007} \bmod 9144407 = 050008.$$
$$m_4 = c^d \bmod n = 7530290^{7449007} \bmod 9144407 = 040020$$

4.5.4 Schéma de signature numérique RSA

Dans la figure 4.8, Aicha utilise d'abord une fonction de hachage unidirectionnelle convenue pour créer le résumé à partir du message m, $H(m) = D$. Aicha signe, en chiffrant le résumé D, $D^d \bmod n = S$. Le message m et la signature S sont envoyés à Brahim. Brahim reçoit le message et la signature S. Il utilise les clés publiques pour obtenir le résumé $D' = S^e \bmod n$. Il applique ensuite l'algorithme de hachage au message reçu de l'expéditeur pour obtenir $D = H(m)$ et compare D et D'. S'ils concordent, le message est accepté, sinon rejeté.

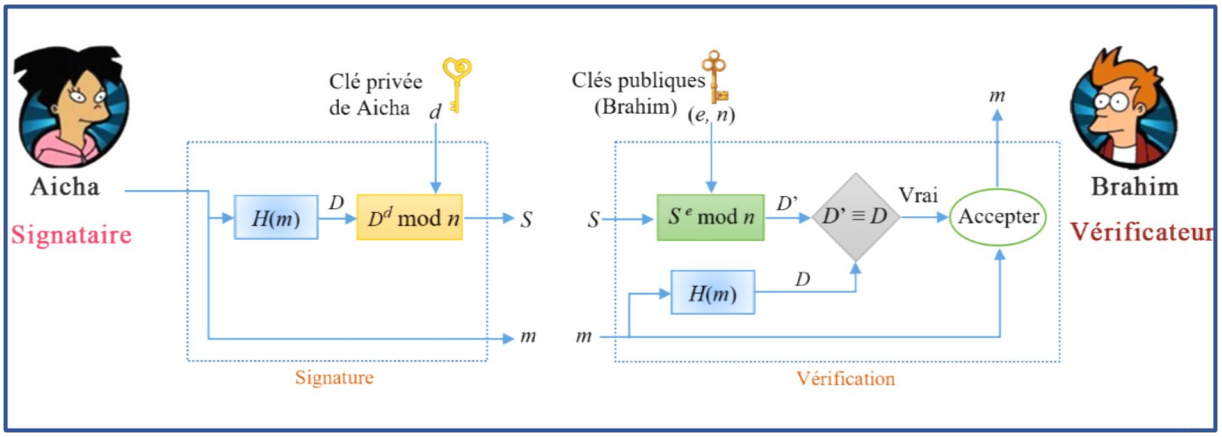

Figure 4.8 : Les étapes de la signature numérique de RSA

4.5.4.1 Exemple de la signature numérique de RSA

Nous illustrons le schéma de signature numérique RSA avec un petit exemple numérique. Bien sûr, cet exemple n'est pas sécurisé, car les nombres sont très petits qu'il serait facile pour qu'un adversaire de factoriser le module n. Les implémentations sécurisées de RSA utilisent des modules n avec des centaines de chiffres.

- ***Création de clé de signature RSA***
  - ✓ Aicha choisit deux nombres premiers secrets $p = 1223$ et $q = 1987$ et calcule son module public $n = p \cdot q = 1223 \cdot 1987 = 2430101$
  - ✓ Aicha choisit une clé publique $e = 948047$ avec la propriété pgcd $(e, \varphi(n))$ = pgcd (948047, 2426892) = 1.
  - ✓ Aicha calcule sa clé de signature privée d en utilisant les valeurs secrètes de p et q pour calculer $\varphi(n) = (p - 1)(q - 1) = 1222 \cdot 1986 = 2426892$ puis elle résoudre la congruence $ed \equiv 1 \bmod \varphi(n)$, $948047 \cdot d \equiv 1 \bmod 2426892$. Elle constate que $d = 1051235$.
- ***Signature RSA***
  - ✓ Aicha sélectionne un message numérique m à signer, elle calcul son empreinte $D = H(m) = 1070777$ avec $1 \leq D < n$.
    Elle calcule la signature numérique $S \equiv D^d \bmod n$, $S \equiv (1070777)^{1051235} \equiv 153337 \bmod 2430101$.
  - ✓ Aicha envoie l'empreinte du message m et sa signature : $D = 1070777$ et $S = 153337$.
- ***Vérification RSA***
  - ✓ Brahim utilise les clés publiques n et e pour calculer $S^e \bmod n = D'$, $153337^{948047} \equiv 1070777 \bmod 2430101$, d'où $D' = 1070777$.
  - ✓ Brahim vérifie que la valeur de D'est égale $D = 1070777$, il accepte le message.

4.6 Signatures sans certificat basé sur RSA

La cryptographie de signature sans certificat est une approche efficace largement étudiée, car elle élimine le besoin d'autorité de certification (CA) dans l'infrastructure à clé publique (PKI : *public key infrastructure*).

Il existe de nombreuses théories proposées pour une transmission sécurisée. À l'heure actuelle, la plupart des théories sont sur papier mais loin de la véritable application. RSA a déjà été implémenté dans diverses applications comme RCSF, le cloud computing, etc.

Les principaux avantages du schéma RSA-CLS (Certificate Less Signature) basé sur RSA sont d'éviter les opérations de couplage, car elles sont coûteuses pour le RCSF à ressources limitées.

Ainsi, dans notre contribution, nous avons proposé un schéma de signature sans certificat basé sur RSA, en particulier pour les réseaux de capteurs sans fil. La complexité de la sécurité de notre contribution est étroitement liée aux définitions données ci-dessous :

Nous examinons brièvement la définition du problème de l'hypothèse forte RSA (*Strong RSA Assumption*). Et le problème du logarithme discret (DLP : *Discrete Logarithm Problem*) [68] :

Définition 1 (l'hypothèse forte de RSA) : Soit $n = p.q$ un module de type RSA et soit G un sous-groupe cyclique de $\mathbb{Z}_n^*$ d'ordre $\#G$, $\lceil \log_2 (\#G) \rceil = l_G$. Étant donné (n, e) et $y \in G$, l'hypothèse forte de RSA consiste à trouver $y \in \mathbb{Z}_n^*$ satisfait $y^e \bmod n = z$.

La sécurité de notre schéma sera réduite à la dureté du problème du logarithme discret (DL) dans le groupe dans lequel la signature est construite.

Définition 2 (Le problème du logarithme discret) : Soit $n = p.q$ un nombre modulaire RSA satisfait $p = 2p' + 1$, $q = 2q' + 1$, et $g \in \mathbb{Z}_n^*$ est un générateur d'ordre $p'q'$, pour des éléments donnés g, y, n, son but est de calculer l'exposant x tel que $y = g^x \bmod n$.

4.6.1. Modèle de réseau

Dans ce scénario, nous utiliserons un réseau sans fil homogène comme le montre la figure

4.9. Certaines hypothèses seront les suivantes :

1) Tous les nœuds de capteurs seront de nature homogène.

2) La station de base fonctionnera en tant que KGC (Key Generation Center) et vérificateur car chaque communication sera entre les nœuds de capteur et la station de base.

3) La station de base aura une énergie illimitée.

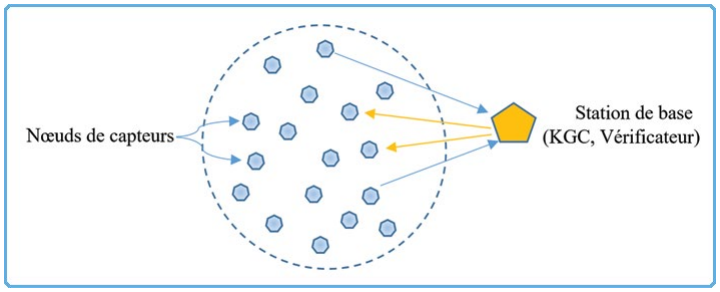

Figure 4.9 : Modèle de réseau du schéma proposé

4.6.2 Les étapes de signature sans certificat

Un schéma de signature sans certificat est défini par sept algorithmes :

1. **Configuration :** Cet algorithme est exécuté par le KGC (Key Generation Center) pour initialiser le système. Il prend en entrée un paramètre de sécurité 1^k et sort une liste de paramètres *params* système et la clé secrète principale d. Les paramètres système

params sont publics, par contre la clé secrète principale *d* est connue uniquement par KGC.

2. **Extraction de clé privée partielle :** Cet algorithme est exécuté par le KGC. Il prend en entrée les paramètres système *params*, la clé secrète principale *d* et un identité $ID \in \{0,1\}^*$. Il génère la clé privée partielle d_{ID}, envoyée à l'utilisateur via un canal sécurisé.
3. **Définir la valeur secrète :** Il s'agit d'un algorithme probabiliste, géré par l'utilisateur. Il prend les paramètres système *params* et l'identité de l'utilisateur *ID* comme entrée et génère une valeur secrète x_{ID}.
4. **Définir la clé privée :** Il s'agit d'un algorithme déterministe, géré par l'utilisateur. Il prend les paramètres du système *params*, une clé privée partielle d_{ID} et une valeur secrète x_{ID} comme entrées et sorties d'une clé privée complète $SK_{ID.}$
5. **Définir la clé publique :** Il s'agit d'un algorithme déterministe, géré par l'utilisateur. Il prend les paramètres système *params*, l'identité de l'utilisateur *ID*, la clé privée partielle d_{ID} et la valeur secrète x_{ID} comme entrées et sorties d'une clé publique $PK_{ID.}$
6. **Signature sans certificat :** Cet algorithme est exécuté par l'utilisateur. Il prend les paramètres du système *params*, l'identité de l'utilisateur *ID,* la clé privée SK_{ID} et un message *m* en entrée. Il génère une signature *S* sans certificat correcte du message *m*.
7. **Vérification de la signature :** Cet algorithme est exécuté par l'utilisateur. Il prend les paramètres du système *params*, l'identité de l'utilisateur *ID*, la clé publique PK_{ID}, le message *m* et la signature *S* en entrée et renvoie *true* si la signature est correcte, sinon *false.*

4.6.3 Proposition de schéma de signature sans certificat basé sur RSA

1. **Configuration :** Étant donné un paramètre de sécurité 1^k en entrée dans KGC, un groupe RSA (n, p, q, e, d) est généré, où p' et q' sont deux grands nombres premiers qui satisfont $p = 2p' + 1$ et $q = 2q' + 1$, $n = pq$ est un nombre modulaire RSA, $e < \varphi(n)$ est la clé publique de (KGC) et satisfait gcd (e, $\varphi(n)$) = 1 et $ed \equiv 1 \bmod \varphi(n)$.
2. Choisissons deux fonctions de hachage cryptographiques H et H_0 qui satisfont :
   H_0: $\{0,1\}^* \rightarrow \mathbb{Z}_n^*$ et H: $\mathbb{Z}_n^4 \times \{0,1\}^* \rightarrow \{0,1\}^l$, où l est un paramètre de sécurité. La clé secrète principale est d et les paramètres publics du système sont $params = \{n, e, H, H_0\}$.
3. **Extraction de clé privée partielle :** Pour un utilisateur avec un $ID \in \{0,1\}^*$; KGC calcule la clé privée partielle en utilisant la clé secrète principale : $d_{ID} = H_0(ID)^d \bmod n$.
4. **Définir la valeur secrète :** Étant donné les paramètres *params* et l'identité *ID*, l'utilisateur choisit au hasard $x_{ID} \in \mathbb{Z}_{2^{|n|/2-1}}$, où $|n|$ désigne la taille binaire de *n.*
5. **Définir la clé privée :** Étant donné la clé privée partielle d_{ID}, la valeur secrète x_{ID} *et l'* l'identité *ID* d'un utilisateur, la sortie $SK_{ID} = (x_{ID}, d_{ID})$.
6. **Définir la clé publique :** Tenir compte de la clé privée partielle d_{ID}, la valeur secrète x_{ID} et l'identité d'utilisateur *ID*, la sortie $PK_{ID} = H_0(ID)^{x_{ID}} \bmod n$.
7. **Signature sans certificat :** Étant donné un message *m*, les paramètres système *params*. L'utilisateur avec l'identité *ID* calcule les étapes suivantes, en utilisant sa clé privée :

   7.1) Choisi au hasard deux nombres r_1, $r_2 \in \mathbb{Z}_{2|n|/2-1}$.

   7.2) Calcule $R_1 = (x_{ID})^e H_0(ID)^{r1} \bmod n$ et $R_2 = H_0(ID)^{r2} \bmod n$.

   7.3) Calcule $h = H(R_1, R_2, ID, PK_{ID}, m)$.

7.4) Définit $u_1 = x_{ID}.(d_{ID})^{r1-h} \bmod n$ et $u_2 = r_2 - x_{ID}.h$.

Enfin, la signature sans certificat résultante du message m est $S = (u_1, u_2, h)$.

8. **Vérification de la signature :** A la réception de la signature sans certificat $S = (u_1, u_2, h)$ du message m, le vérificateur s'exécute comme suit :
   1) Calcule $R'_1 = u_1^e \, H_0(ID)^h \bmod n$ et $R'_2 = H_0(ID)^{u2}(PK_{ID})^h \bmod n$.
   2) Accepte, si et seulement si l'équation suivante contient $h = H(u_1^e \, H_0(ID)^h \bmod n, H_0(ID)^{u2} PK_{\mathrm{ID}}{}^h \bmod n, ID, PK_{ID}, m)$.

Exactitude :

Dans ce qui suit, nous montrons que notre schéma est correct et satisfait à l'exhaustivité.

$H(u_1^e \, H_0(ID)^h \bmod n, H_0(ID)^{u2} PK_{\mathrm{ID}}{}^h \bmod n, ID, PK_{ID}, m)$

$= H((x_{ID}.(d_{ID}{}^{r1-h})^e \, H_0(ID)^h \bmod n, \; H_0(ID)^{r2-(xID)h} \, PK_{\mathrm{ID}}{}^h \bmod n, ID, PK_{ID}, m)$

$= H((x_{ID})^e \, H_0(ID)^{(r1-h)ed} \, H_0(ID)^h \bmod n, H_0(ID)^{r2-(xID)h+(xID)h} \bmod n, ID, PK_{ID}, m)$

$= H((x_{ID})^e \, H_0(ID)^{r1} \bmod n, H_0(ID)^{r2} \bmod n, ID, PK_{ID}, m) = H(R_1, R_2, ID, PK_{ID}, m) = h.$

4.6.4 Types des adversaires et leurs comportements de la signature sans certificat

4.6.4.1 Le modèle d'oracle aléatoire

En cryptographie, un oracle aléatoire est un oracle (une boîte noire théorique) qui répond à chaque requête unique avec une réponse (vraiment) aléatoire choisie uniformément dans son domaine de sortie, afin de fournir des « preuves de sécurité » pour certains protocoles cryptographiques, en particulier les signatures numériques.

Le modèle d'oracle a été introduit, en 1993, par les cryptologues Mihir Bellare et Phillip Rogaway [74]. Ces oracles sont très utilisés dans les schémas où de fortes hypothèses de caractère aléatoire sont nécessaires pour la sortie de la fonction de hachage.

Il s'agit, généralement, d'une fonction de hachage modélisée par un oracle aléatoire. De manière informelle, cela signifie que l'on considère la fonction de hachage H comme une boîte noire qui répond à une requête pour la valeur de hachage d'une chaîne de bits m en donnant une valeur aléatoire. Pour chaque requête q, l'oracle fait un choix aléatoire indépendant, sauf qu'il tient un registre de ses réponses $H(m)$ et répète la même réponse si m est interrogé à nouveau :

- Pour une requête q jamais formulée auparavant, l'oracle répond une chaîne de bits $s_q \in \{0,1\}^n$
- Pour une requête déjà reçue, l'oracle renvoie s_q, si l'oracle n'enregistre pas ses réponses, sinon il répond $s_{q'}$ différent de s_q

Dire que $H(.)$ peut être modélisé par un oracle aléatoire est un argument beaucoup plus fort, plus que la résistance à la collision, et la résistance à la pré-image.

4.7 Analyse de sécurité

En ce qui concerne le modèle de sécurité, le schéma de signature sans certificat est différent des schémas de sécurité ordinaires. Il existe deux types d'adversaires (de type I et de type II) auxquels les systèmes sans certificat sont vulnérables. L'adversaire A_1 de type I représente un nœud de capteur externe malveillant qui attaque le schéma CLS.

A_1 n'est pas autorisé à accéder à la clé principale SK_{ID}, mais A_1 peut donc modifier les clés publiques PK_{ID} du nœud afin d'attaquer le schéma CLS.

L'adversaire A_2 de Type II montre un centre de génération de clés KGC corrompu. A_2 peut donc générer une clé privée SK_{ID} pour un utilisateur mais il ne peut pas changer la clé publique PK_{ID} d'un nœud.

Pour appeler un schéma CLS sécurisé, il devrait fournir une prévisibilité contre les attaques adaptatives à un message choisi et à une identité dans le modèle accusatoire de la cryptographie sans certificat qui se compose des deux types adversaires indiqués ci-dessus :

Dans ce qui suit, nous donnerons une analyse de sécurité de notre schéma proposé sous Random Oracle Model et nous montrerons que notre schéma est possible d'empêcher le réseau d'attaques de type I et de type II avec une probabilité non négligeable.

4.7.1 Adversaires A_1

Nous essayons de monter qu'il existe un adversaire A_1 de type I qui peut demander des requêtes de hachage q_{H0} et q_H à des oracles aléatoires H_0 et H, des requêtes de signature q_s, des requêtes d'extraction de clé privée partielle q_{ppk} et des requêtes d'extraction de clé privée q_p, et peuvent rompre notre schéma proposé RSA-CLS dans un temps polynomial τ. A cette fin, A_1 a besoin d'un adversaire B pour résoudre le problème fort de RSA (voir la définition 1 section 3) de sorte que B puisse utiliser A_1 comme boîte noire pour résoudre l'hypothèse forte de RSA. L'objectif est de trouver $y^e = z$, où $y \in \mathbb{Z}_n^*$ avec n module RSA et (n, e, z) est une instance du problème RSA.

Supposons qu'il y ait un adversaire A_1 qui peut changer la clé publique d'un nœud de capteur de l'identité ID et générer une signature malveillante pour le vérificateur. Pour cela, A_1 effectuera les étapes suivantes :

Configuration : l'adversaire B sélectionne deux fonctions de hachage H_0 et H comme oracle aléatoire. d est la clé secrète principale, inconnue par B, qui satisfait $ed \equiv 1 \bmod (n)$. Les paramètres système (e, n) sont publics pour tous. L'adversaire B maintient trois listes H_0-*list*, *H-list* et *Key-list*, qui sont initialement vides. L'adversaire B envoie (e, n, z, H_0, H) en sortie finale à l'adversaire A_1.

Requêtes : A tout moment, A_1 est autorisé à accéder aux oracles plusieurs fois d'une façon polynomiale. Ensuite, B simule les requêtes oracle de A_1 comme suit :

1) **Requêtes *H_0-Hash*** : A_1 peut interroger l'oracle aléatoire H_0 à tout moment avec une identité ID. En réponse à ces requêtes, B retourne une variable V biaisée au hasard $V \in \{0,1\}$ de telle sorte que la probabilité $Pr\ (V = 0) = \rho$.
   Ensuite, B choisit au hasard $t_{ID} \in \mathbb{Z}_n$ et calcule $h^0_{ID} = z_V\ (t_{ID})^e$ et l'envoie à A_1. B ajoute (ID, h^0, t_{ID}, V) à H_0-*list*.

2) **Requêtes *H-Hash* :** A_1 peut interroger l'oracle aléatoire H à tout moment avec $h = H(R_1, R_2, ID, PK_{ID}, m)$. Pour chaque requête $(R_1, R_2, ID, PK_{ID}, m)$, B vérifie d'abord la liste *H-list*:
   2.1) Si $(R_1, R_2, ID, PK_{ID}, m, h)$ existe dans *H-list*, alors B définit $H(R_1, R_2, ID, PK_{ID}, m) = h$ et renvoie h à A_1.
   2.2) Sinon, il choisit aléatoirement $h \in \mathbb{Z}_n$ et ajoute l'enregistrement $(R_1, R_2, ID, PK_{ID}, m, h)$ à *H-list*. B envoie h à A_1 comme une réponse correspondante.

3) **Requêtes d'extraction de la clé privée partielle** : A tout moment, A_1 peut interroger l'oracle en donnant un identifiant ID. B affiche un symbole $\perp$ si l' ID n'a pas été créé. Sinon, si ID a été créé et $V = 0$, alors B retourne t_{ID} à l'adversaire A_1. Sinon, B renvoie l'échec et met fin à la simulation.

4) **Requêtes de clé publique :** A tout moment, A_1 peut interroger l'oracle en donnant un identifiant d'identité *ID*. *B* choisit au hasard $x_{ID} \in \mathbb{Z}_{2|n|/2-1}$ et recherche (ID, h^0_{ID}, t_{ID}, V) dans H_0-*liste*. Ensuite, *B* ajoute (ID, $PK_{ID} = h^0_{ID}{}^{x_{ID}}$, x_{ID}, V) à *Key-List* et envoie PK_{ID} à A_1.

5) **Private-Key-Extract :** Pour un *ID* d'identité donné choisi par A_1, *B* recherche (ID, h^0_{ID}, t_{ID}, *pièce*) dans H_0-*liste*. Si $V = 1$, alors *B* interrompt, sinon, *B* recherche (ID, $PK_{ID} = h^0_{ID}{}^{xID}$, x_{ID}, V) dans la *Key-List*. *B* retourne $SK_{ID} = (x_{ID}, t_{ID})$ à A_1 comme sortie finale.

6) **Requêtes de remplacement de clé publique :** A_1 peut demander une requête pour remplacer la clé publique PK_{ID} d'un capteur d'identité *ID* par une nouvelle clé publique PK'_{ID} choisie par A_1 elle-même. Par conséquent, *B* remplace la clé publique d'origine PK_{ID} par PK'_{ID} si l'*ID* a été créé dans H_0-*liste*. Si non, il affiche $\perp$.

7) **Les requêtes Signature :** Pour chaque requête sur une entrée (m, ID), affichez $\perp$ si l'*ID* n'a pas été interrogé auparavant. Pour toute entrée (m, ID) avec l'*ID* interrogé, *B* recherche (ID, h^0_{ID}, t_{ID}, V) et (ID, PK_{ID}, x_{ID}, V) dans H_0-*liste* et *Key-list*. Si $V = 0$, *B* produit alors une signature sans certificat *S* sur le message *m* par la clé privée retournée (x_{ID}, t_{ID}). Sinon *B*, se calcule comme suit :

   7.1) *B* choisit aléatoirement $u_1 \in \mathbb{Z}_n^*$, $h \in \{0,1\}^l$ et $u_2 \in \mathbb{Z}_{2|n|/2-1}$.

   7.2) Calcule $R_1 = u_1{}^e H_0(ID)^h$ et $R_2 = H_0(ID)^{u2} PK_{ID}{}^h$, où PK_{ID} peut être une clé publique remplacée.

   7.3) *B* recherche si (R_1, R_2, ID, PK_{ID}, m) existe dans *H-liste* . S'il existe, il abandonne. Sinon, *B* définit $H(R_1, R_2, ID, PK_{ID}, m) = h$ et ajoute (R_1, R_2, ID, PK_{ID}, m, h) dans la *H-liste*.

   7.4)La signature résultante $S = (u_1, u_2, h)$ est retournée à A_1.

Sortie : Après toutes les requêtes, A_1 sort ces faux éléments : (ID^*, PK_{ID*}, m^*, $S^* = (u_1{}^*, u_2{}^*, h^*)$), il gagne la partie si les conditions suivantes sont remplies :

1) Si S^* est une signature valide, alors $h^* = H(R_1{}^*, R_2{}^*, PK_{ID*}, ID^*, m)$, qui est dans la *H-liste*, où $R_1{}^* = u_1{}^{*e} H_0 (ID^*)^{h^*}$ et $R_2{}^* = H_0 (ID^*)^{u_2{}^*} PK^{h^*}{}_{ID*}$.
2) $V^* = 1$ de l'enregistrement (ID^*, h^0_{ID*}, t_{ID*}, V^*) dans la H_0-*liste*.

Avant de suivre notre processus de l'attack A_1, On s'arrête à Lemme de Foking [80].

Lemme de Foking :

Soit A une machine de Turing temporelle polynomiale probabiliste, étant donné seulement les données publiques en entrée. Si A peut trouver, avec une probabilité non négligeable, une signature valide (m, S_1, h, S_2), avec une probabilité non négligeable, une relecture de cette machine, avec la même bande aléatoire et un oracle différent, génère deux signatures valides (m, S_1, h, S_2) et (m, S_1, h', S_2') tels que $h \neq h'$.

En appliquant ce Lemme, après avoir rejoué A_1 avec la même bande aléatoire mais différents choix d'oracle *H*, *B* peut obtenir une autre signature sans certificat valide (ID^*, PK_{ID*}, m^*, $S'^* = (u_1'^*, u_1'^*, h'^*)$).

Donc, ils doivent satisfaire $R_1{}^* = (u_1{}^*)^e H_0(ID^*) h^*$ et $R_1{}^* = (u_1'^*)^e H_0 (ID^*) h'^*$. Ainsi, nous avons la relation suivante :

$$(u_1{}^*)^e H_0(ID^*) h^* = (u_1'^*)^e H_0 (ID^*) h'^*$$

$$\left(\frac{u_1{}^*}{u_1{}'^*}\right)^e = H_0\,(ID*)^{h-h'}$$

$$\left(\frac{u_1{}^*}{u_1{}'^*}\right)^e = (zt_{DI*}{}^e)^{h-h'}$$

$$\left(\frac{u_1{}^*}{(t_{DI*})^{h-h'}\,u_1{}'^*}\right)^e = (z)^{h-h'}$$

D'après le lemme de Foking, $h'^* - h^* \neq 0$ et e est un nombre premier. Cela signifie que gcd $(e, h'^* - h^*) = 1$, alors il existe deux nombres a, b satisfaisant $ae + b(h'^* - h^*) = 1$. Ainsi, nous pouvons obtenir :

$$z = (z)^{ae\,+\,b(h'*\,-\,h*)} = (z)^{ae}\,z^{b(h'*\,-\,h*)} = (z)^{ae}\left(\frac{u_1{}^*}{(t_{DI*})^{h-h'}\,u_1{}'^*}\right)^{eb} =$$

$$\left(z^a\left(\frac{u_1{}^*}{(t_{DI*})^{h-h'}\,u_1{}'^*}\right)^b\right)^e = y^e.$$

$$\text{D'où } y = \left(z^a\left(\frac{u_1{}^*}{(t_{DI*})^{h-h'}\,u_1{}'^*}\right)^b\right)$$

Cela montre que B peut résoudre l'hypothèse forte de RSA avec une probabilité Pr, donc c′est une contradiction avec l'hypothèse forte de RSA.

Nous observerons que B ne s'interrompt pas pendant toute la simulation, A_1 peut forger la signature et la signature sans certificat valide (ID^*, $PK_{ID}{}^*$, m^*, $S'^* = (u_1{}'^*, u_1{}'^*, h'^*)$) satisfait

$R_1{}^* = (u_1{}^*)^e H_0\,(ID^*)^{h*}$ et $R_1{}^* = (\mathrm{u}'_1{}^*)^e H_0\,(ID^*)^{h'*}$.

4.7.2 Adversaires A_2

Théorème. *Dans le modèle d'oracle aléatoire, s'il existe un adversaire de type A_2, qui est autorisé à demander au plus des Hash-requêtes q_{H0}, q_H aux oracles aléatoires H_0 et H, respectivement, et q_s Sign-requêtes, peut briser le schéma de la signature sans certificat proposé avec une probabilité E dans un temps τ, alors il existe un autre algorithme C qui peut utiliser A_2 pour résoudre le problème de logarithme discret.*

Preuve : Supposons qu'il existe un adversaire de type A_2 qui peut casser notre schéma RSA-CLS proposé. Nous allons développer un adversaire C qui utilisera l'adversaire A_2 pour résoudre le problème de logarithme discret (Décrit clairement dans la définition 2 de la section 4.6). Le but est de trouver x qui satisfait $y = g^x \bmod$ et $g \in \mathbb{Z}_n{}^*$. Pour résoudre ce problème en temps polynomial, C doit simuler un défi sur le modèle d'oracle aléatoire et les oracles (oracle de hachage, oracle de génération de clés et oracle de signature) pour A_2. C exécute de la manière suivante :

Configuration : C conserve trois listes *H-list*, H_0-*lis*t et *Key-list* qui sont toutes initialement vides. Soit (e, n) les paramètres du système fournis à l'adversaire. d est la clé secrète principale qui satisfait $ed \equiv 1 \bmod \varphi(n)$, et C connaît les valeurs d et (p, q). Sélectionnons deux fonctions de hachage H et H_0 comme oracles aléatoires. Supposons que $PK_{ID}{}^* = y$ la clé publique d'un utilisateur U^* contestant le schéma et l'ID^* étant l'identité de U^*. Enfin, C envoie une chaîne binaire de paramètres publics (e, d, n, g, H, H_0) à l'adversaire A_2.

Requêtes : L'adversaire A_2 est autorisé à accéder aux fonctions de hachage en tant qu'oracles donnés ci-dessous en temps polynomial. Toutes les simulations de ces oracles sont effectuées par C.

1) **Requêtes *H-Hash* :** Au cours de ce processus de requête d'oracle, A_2 est en mesure de demander un nombre de requêtes q_H au maximum. Pour chaque requête (R_1, R_2, ID, PK_{ID}, m), C choisit aléatoirement $k_{ID} \in \{0,1\}^l$ et définit $k_{ID} = H(R_1, R_2, ID, PK_{ID}, m)$. Enfin, il renvoie k_{ID} à A_2 et ajoute la valeur (R_1, R_2, ID, PK_{ID}, m, k_{ID}) à la *H-liste*.

2) **Requêtes *H_0-Hash* :** Pour cette requête d'oracle, C choisit au hasard $t_{ID} \in \varphi(n)$ et définit $H_0(ID) = g^{tID}$ et le renvoie à l'adversaire A_2 où $\varphi(n)$ est connu comme la fonction d'Euler obtenue par p, q. Enfin, il ajoute (ID, $H_0(ID)$, t_{ID}) à la *H_0-liste*.

3) **Requêtes de clé publique :** A tout moment, A_2 peut interroger l'oracle en lui attribuant une identité ID. Si $ID \neq ID^*$, C choisit au hasard $x_{ID} \in \varphi(n)$ pour calculer $PK_{ID} = H_0(ID)^{xID}$. Puis, il ajoute (ID, $PK_{ID} = H_0(ID)^{xID}$, x_{ID}) à *Key-List*. Sinon, C recherche dans la *H_0-liste* (ID^*, $H_0(ID^*)$, $t_{ID}{}^*$) et calcule $PK_{ID*} = y^{tID*}$. En outre, il ajoute l'enregistrement (ID^*, PK_{ID*}, $\perp$) à *Key-List*. Ensuite, il envoie PK_{ID} à A_2.

4) **Requêtes d'extraction de la clé privée :** Lorsque A_2 génère une clé privée avec une requête d'identité ID, si $ID \neq ID^*$, C recherche d'abord un enregistrement disponible (ID, PK_{ID}, x_{ID}) dans *Key-list* et calcule $d_{ID} = H_0(ID)^d$. Si $ID = ID^*$, alors C interrompt l'oracle.

5) **Signature Oracle :** Pour chaque requête en entrée (m, ID), si $ID \neq ID^*$, alors C obtient d'abord la clé privée associée à l'ID par les **Requêtes d'extraction de clé privée** sur l'ID, puis génère une signature à l'aide de cette clé privée. Si $ID = ID^*$, alors C calcule ce qui suit:

   5.1) Choisi aléatoirement $u_1 \in \mathbb{Z}_n$ et $h \in \{0,1\}^l$, $u_2 \in \mathbb{Z}_{\varphi(n)}$.

   5.2) Calcule $R_1 = u_1{}^e H_0(ID)^h$ et $R_2 = H_0(ID)^{u_2} (PK_{ID})^h$.

   5.3) Recherche si (R_1, R_2, ID, PK_{ID}, m) existe dans la *H-liste*. S'il existe, C abandonne. Sinon, C définit $H(R_1, R_2, ID, PK_{ID}, m) = h$ et ajoute $H(R_1, R_2, ID, PK_{ID}, m, h)$ dans la *H-liste*.

   5.4) La signature résultante $S = (u_1, u_2, h)$ est renvoyée à A_2.

Sortie : Après avoir simulé toutes les requêtes, C génère une contrefaçon :

(ID^*, PK_{ID*}, m^*, $S^* = (u_1{}^*, u_2{}^*, h^*)$) et il gagne ce jeu s'il satisfait aux conditions suivantes :

1) Si S^* est une contrefaçon valide, alors $h^* = H(R_1{}^*, R_1{}^*, PK_{ID*}, ID^*, m)$ qui est dans *H-liste*, où $R_1{}^* = (u_1{}^*)^e H_0(ID^*)^{h*}$ et $R_2{}^* = H_0(ID^*)^{u_2*} (PK_{ID*})^{h*}$.

2) ID^* est l'identité du challenger et $H_0(.)$ est l'oracle interrogé par ID^*.

Lorsque C joue à nouveau la même bande d'oracle aléatoire mais avec différents choix d'oracle H, selon le lemme de (Forking) [80], C peut générer une autre signature sans certificat valide : (ID^*, PK_{ID*}, m^*, $S^{*\prime} = (u_1{}^{*\prime}, u_2{}^{*\prime}, h^{*\prime})$) alors ceux-ci devraient rester vrais :

$R_2{}^* = H_0(ID^*)^{u_2} (PK_{ID*})^{h*}$ et $R_2{}^* = H_0(ID^*)^{u_2*'} (PK_{ID*})^{h*'}$

$$H_0(ID^*)^{u_2} (PK_{ID*})^{h*} = H_0(ID^*)^{u_2*'} (PK_{ID*})^{h*'}$$

$$H_0(ID^*)^{u_2 - u_2*'} = (PK_{ID*})^{h*' - h*}$$

$$(g)^{tID*(u_2 - u_2*')} = y$$

$$(g)^{tID*(u_2 - u_2*') / (h*' - h)} = y$$

Nous avons ainsi le logarithme discret de y à la base de g est $t_{ID*}(u_2 - u_2*') / (h*' - h)$, indiquant que le problème discret peut être résolu par C. Évidemment, c'est une contradiction à la difficulté de résoudre le problème du logarithme discret.

Définition 3 : S'il n'y a pas d'attaquant à temps polynomial $A \in \{A_1, A_2\}$ qui peut gagner dans les deux jeux ci-dessus avec une probabilité non négligeable, alors un schéma CLS est censé être existentiellement infalsifiable sous les attaques adaptatives à message choisi.

D'après la définition 1, les deux types d'adversaires A_1et A_2 considérés dans la cryptographie sans certificat peuvent forger les signatures valides au nom de n'importe quel utilisateur sur n'importe quel message de leur choix, soit avec une probabilité non négligeable (gagnons) ou faible (perdons), et avec des conditions précisées dans la sortie des simulations de chaque type d'adversaires.

Outre de ces attaques, notre schéma conserve également les propriétés de sécurité d'origine de la signature, à savoir l'intégrité, l'authentification et la non-répudiation. Dans ce qui suit, nous montrerons que notre système garantit ces éléments essentiels de sécurité.

4.7.3 Intégrité

Dans notre schéma proposé, la station de base SB vérifie l'intégrité du message m en vérifiant la signature $S = (u_1, u_2, h)$, où $h = (R_1, R_2, ID, PK_{ID}, m)$.

Afin de vérifier le message, SB utilise la clé publique du signataire et la clé publique e de KGC pour calculer les valeurs R_1' et R_2' en premier. Le vérificateur utilise ensuite ces valeurs, la clé publique du signataire et le message m pour générer $h' = H(R_1', R_2', ID, PK_{ID}, m)$. Lorsque le vérificateur passe à comparer h'et h et la vérification de la signature S, il trouve que le message reçu m est égal à la valeur m dans la signature S. Par conséquent, notre schéma proposé fournit un mécanisme pour convaincre que le message et la signature transmis sont corrects et complets. Les détails de la vérification de la signature et l'exactitude de l'équation h'et h sont décrits dans la section 3.2 (phase de vérification/exactitude).

4.7.4 Non-répudiation

La non-répudiation est un cas où un signataire malveillant, qui génère une signature avec le message m, mais refuse ensuite la signature. Dans notre schéma proposé lorsqu'un signataire génère une signature, il utilise son ID pour générer la signature correcte. Pendant la phase de signature, le nœud calcule les valeurs de R_1, R_2, où il utilise sa valeur d'identité ID fourni par KGC. Lorsque la signature $S = (u_1, u_2, h)$ est générée, nous pouvons voir que $h = (R_1, R_2, ID, PK_{ID}, m)$ contient l'identité ID. Ainsi, au cours de la phase de vérification, le vérificateur peut facilement découvrir que cette signature est générée par le nœud avec l'identité ID. Le signataire doit appliquer la fonction de hachage unidirectionnelle H_0 sur l'ID pour générer la signature, donc sans ID correct, il ne peut pas produire la signature correcte. Le schéma proposé peut donc empêcher le signataire de répudier sa signature.

4.7.5 Les falsifications

Un attaquant peut appliquer la falsification de 2 manières : la falsification du message et la falsification du message et de la signature.

4.7.5.1 Falsification du message :

Supposons qu'un attaquant est capable d'intercepter la signature $S = (u_1, u_2, h)$ et le message m et modifie le message m en m'. Lorsqu'il envoie la signature $S = (u_1, u_2, h)$ et le message m'

au vérificateur, pour vérifier le message et la signature, le vérificateur utilise la clé publique du signataire PK_{ID} et sa propre clé publique maître e pour calculer $R_1' = u_1{}^e H_0(ID)^h \bmod n$ et $R_2' = H_0(ID)^{u2}(PK_{ID})^h \bmod n$. le vérificateur utilise ensuite ces valeurs pour générer $h = (R_1', R_2', ID, PK_{ID}, m')$ et vérifie si h est égal à h'. Dans ce cas, ne sera pas possible car il y a une altération du message, donc le vérificateur peut facilement détecter que quelque chose ne va pas avec le message et rejettera le message.

4.7.5.2 Falsification du message et de la signature

Supposons maintenant le cas où l'attaquant intercepte la signature $S = (u_1, u_2, h)$ et le message m et modifie à la fois le message et la signature comme $S' = (u_1', u_2', h')$ et m'. L'attaquant envoi S'et m' au vérificateur. Mais comme nous pouvons remarquer que h' contient R_1, R_2, ID, PK_{ID} et m, les valeurs u_1 et u_2 ne peuvent pas être calculées sans PK_{ID} correct et la clé secrète principale d. Donc, sans connaissance correcte de ces valeurs, l'attaquant ne peut pas passer la vérification.

Comme nous pouvons le voir par la démonstration donnée ci-dessus, notre système peut résister à l'attaque par contrefaçon dans un réseau de capteurs sans fil.

4.7.6 Analyse et performance

Comme nous le savons, la technique RSA est appliquée dans différentes atmosphères depuis des décennies. Jianhong Zang *et al.* ont proposé un schéma sans certificat basé sur RSA dans le cadre d'un problème de RSA fort et de logarithme discret [73]. Nous avons comparé son schéma avec le nôtre sur les propriétés de sécurité de la signature sans certificat basé sur RSA (voir le tableau 4.4).

Schémas	Nombre de phases dans l'algorithme	Sécurité contre les attaques de type I	Sécurité contre les attaques de type II	Authentification	Non-répudiation	Intégrité
Zang et al. [73]	7	Non	Oui	Oui	Oui	Oui
Le schéma proposé	7	Oui	Oui	Oui	Oui	Oui

Tableau 4.4 : Comparaison des propriétés de sécurité de notre schéma avec Zang *et al.*

L'inconvénient du schéma de [73] c'est qu'il n'était pas sécurisé sous une attaque de type I.

Donc, notre schéma offre une sécurité contre toutes les menaces de sécurité mentionnées. Nous pouvons donc affirmer que notre algorithme fonctionne mieux que d'autres schémas dans le cas des réseaux de capteurs sans fil.

4. Conclusion

Lorsque nous discutons du schéma de signature sans certificat, nous voyons que la plupart des schémas de signature sans certificat existants ont été produits sur la base d'une technique d'appariement bilinéaire qui est très difficile à construire dans un environnement réel. Ainsi, dans la contribution présentée dans ce chapitre, nous avons proposé un schéma de signature sans certificat basé sur RSA pour les réseaux de capteurs sans fil qui peut être mis en œuvre de manière efficace. Au niveau de la sécurité de notre schéma RSA-CLS, nous avons prouvé qu'il était sûr sous certaines conditions bien étudiées contre les adversaires de type A_1 et A_2. Notre

schéma utilise 7 algorithmes temporels polynomiaux pour la génération de signatures, et dispose de meilleures performances que Zang *et al.* [73] dans un réseau de capteurs sans fil et fournissent presque tous les objectifs de sécurité essentiels.

Chapitre

5 Vers une approche d'entités nommées de la langue amazighe pour les réseaux de capteurs sans fils

5.1 Introduction :

L'avènement du réseau sans fil de la cinquième génération (5G) et sa convergence avec les applications verticales constituent le fondement de la future société connectée qui devrait prendre en charge 125 milliards d'appareils d'ici 2030 (IHS Markit)[98]. Comme ces applications et appareils sont caractérisés par des exigences de connectivité omniprésentes, les futurs réseaux 5G et au-delà deviennent de plus en plus complexes. Outre l'augmentation de la complexité des stations de base (BS) et des équipements utilisateur (UE), des défis importants surgissent de la planification initiale du réseau aux phases de déploiement et d'exploitation et de gestion dépendant de la situation [99].

De nombreuses applications MTC (Machine-Type-Communication) futures prises en charge par la 5G et au-delà nécessiteront les réseaux sans fil sous-jacents pour atteindre une haute disponibilité, fiabilité et sécurité, un temps de transit très court et une faible latence [100].

En outre, un certain nombre de nouveaux cas d'utilisation immersifs MTC gourmands en données apparaîtront, y compris les wearables (technologie portable : vêtements, lunettes..), les réalités virtuelles, les produits intelligents et les systèmes de support où la plupart d'entre eux utiliseront une infrastructure de données backend intégrée et un moteur d'analyse pour fournir des services contextuels. Tout cela nécessite que le réseau de nouvelle génération (5G) adopte une approche intelligente et contextuelle pour la planification, la conception, l'analyse et l'optimisation du réseau [101].

Les humains communiquent avec des appareils intelligents IoT(Internet of Things) tels que Alexa et Siri en langage naturel; ainsi, le traitement du langage naturel (TLN) devient une partie intégrante des dispositifs IoT [102–103]. Pour réaliser l'interaction simple entre l'homme et l'ordinateur, il est préférable que les machines comprennent les langages naturels. Afin d'incorporer des fonctionnalités de compréhension du langage naturel dans les machines, plusieurs techniques de TLN sont utilisées. Les techniques TLN décrivent comment les machines peuvent être utilisées pour traiter et analyser les langues naturelles.

Au cours des années précédentes, plusieurs applications TLN telles que la synthèse de texte, la réponse aux questions, la traduction automatique et les chatbots (aussi nommé dialogueur ou agent conversationnel, agent qui dialogue avec un utilisateur) conversationnels intelligents ont été développées avec la capacité de comprendre le langage naturel [104].

Au cours du développement de ces applications TLN, la tâche de Reconnaissance d'Entité Nommée (NER) est appliquée en tant qu'étape de prétraitement pour améliorer les performances du système global.

Il existe plusieurs éléments technologiques existantes à partir desquelles commence les technologies sémantiques fournissent un moyen de surmonter le problème d'intégration [105], où les réseaux de capteurs orientés vers l'ingénierie présentent des limites sur un système

complexe à grande échelle. Il existe deux normes sémantiques proposées pour modéliser les dispositifs capteurs : le réseau de capteurs sémantiques SSN (Semantic Sensor Network) du World Wide Web Consortium (W3C) et Sensor Web Enablement (SWE) de l'Open Geospatial Consortium (OGC). Bien que ces normes fournissent des interfaces utiles pour un accès omniprésent aux données, elles ne fournissent pas une interface de haut niveau facile à utiliser pour interroger et fusionner les données de réseaux hétérogènes. Par exemple, le Resource Description Framework (RDF) dans le protocole W3C et SPARQL et le langage de requête RDF (SPARQL) dans SSN permettent à la fois la sélection et la fusion des capteurs, mais l'utilisation de ce système peut être difficile en raison des langages de requête spécifiques à l'application utilisés. Cependant, l'avancée du traitement du langage naturel facilite la compréhension de la signification sémantique d'un texte. TLN peut nous aider à identifier les principaux objets auxquels les entrées s'adressent et à quel type d'objets elles se réfèrent. La TLN a été appliquée à de nombreux domaines d'application tels que la reconnaissance vocale et la réponse aux questions. Cependant, afin d'effectuer des recherches omniprésentes sur les données des capteurs, nous avons besoin d'un système qui reliera la technologie TNL et les systèmes de capteurs.

La contribution de ce travail est l'extension de TLN au domaine du réseau de capteurs pour répondre à des requêtes de haut niveau qui seraient traditionnellement exprimées dans un langage de requête spécifique à l'application. Les composants de requête en langage naturel sont liés au Web sémantique et sont donc capables de s'adapter aux hétérogénéités présentes dans les réseaux de capteurs sous-jacents, simplifiant considérablement l'accès aux données de capteurs omniprésents. La Reconnaissance d'Entité Nommée (REN) est une sous-tâche de l'activité d'extraction d'information dans des corpus documentaires. Elle consiste à rechercher des objets textuels (c'est-à-dire un mot, ou un groupe de mots) catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, articles de presse, des rapports et même des tweets, etc.

Grâce à la disponibilité de corpus annotés, les méthodes d'apprentissage supervisé ont été largement adoptées et prévalent celles qui ne sont pas supervisées. De tels systèmes NER de pointe ont atteint des performances aussi élevées que les annotateurs humains. De leur côté, les systèmes REN s'améliorent avec l'avantage de plus d'annotations des corpus pour apprendre.

Les méthodes traditionnelles de lutte contre les RENs vont de l'appariement par dictionnaire, des règles heuristiques, à l'étiquetage de séquence basé sur les modèles de Markov cachés (HMM) / champs conditionnels aléatoires (CRF : Conditional Random Fields). Les deux premières approches ne nécessitent pas de données de formation, mais impliquent généralement des règles et des hypothèses ad hoc qui peuvent limiter le type d'entités et de textes auxquels elles pourraient s'appliquer. Les étiqueteurs basés sur CRF ont donné des performances élevées dans les tâches d'apprentissage de séquence et sont à la pointe de la technologie pour certaines tâches de reconnaissance d'entité. Cependant, la nature supervisée du CRF implique une quantité assez importante de données d'entraînement qui doivent être annotées par l'homme. Par conséquent, il n'est applicable que dans un nombre limité de paramètres.

Les meilleurs systèmes REN pour l'anglais produisent une précision presque humaine. Un tel système peut le faire à un niveau de précision d'environ 93,39%, alors qu'un humain atteindrait une précision d'environ 97%. Cependant, la langue amazighe présente des défis uniques à surmonter.

Ce travail explique le besoin de reconnaissance en EN pour la langue amazighe et discute des problèmes et des défis liés aux tâches de reconnaissance en EN pour la langue amazighe. Il explore également diverses méthodes et techniques qui sont utiles pour la création de ressources

d'apprentissage et de lexiques qui sont importants pour l'extraction d'EN à partir d'un langage naturel non structuré.

De plus, l'application de REN à l'amazighe est considéré comme crucial, il aide à améliorer les performances du traitement du langage naturel (TLN) en langue amazighe. Par exemple, lors de l'exécution de tâches liées à la gestion de grandes quantités d'informations, les systèmes REN peuvent aider dans les tâches d'extraction d'informations (EI), de récupération d'informations (RI) et de réponse aux questions (RQ).

Ce document est organisé comme suit : La section 2 met en évidence certains travaux connexes, la section 3 traite le langage naturel avec l'ontologie et leurs annotateurs temporelle, spatiale et caractéristique, la section 4 présente le traitement du langage naturel (TLN) et son interaction homme-machine et la suppression des mots (données), la section 5 couvre une précision (définition) du traitement automatique des langues naturelles (TALN), une brève description de la langue amazighe, sa graphie Tifinaghe Unicode**,** et sa morphologie. Les sections 6, 7 et 8 fournissent les défis et les objectifs de la reconnaissance des nouvelles entités NER de l'amazighe et son corpus de test, la section 9 couvre le système REN typique, notre approche hybride basé sur le modèle CRF (Conditional Random Fields) et les algorithmes utilisés pour l'identification des noms et les RENs, la section 10 rassemble quatre répertoires géographiques des listes de EN différentes fabriqués manuellement et ses mot déclencheurs, les résultats et discussions sont présentés dans la section 11, La section 12 conclut ce chapitre et parle des travaux futurs.

5.2 TRAVAUX CONNEXES

CASSARAM : Un modèle de recherche, de sélection et de classement des capteurs contextuel pour l'Internet des objets pour relever les défis de recherche liés à la sélection des capteurs lorsqu'un grand nombre de capteurs avec des fonctionnalités qui se chevauchent et parfois redondantes sont disponibles. Ce modèle propose la recherche et la sélection de capteurs en fonction des priorités des utilisateurs. CASSARAM considère un large éventail de caractéristiques des capteurs pour la recherche tels que la fiabilité, la précision, la durée de vie de la batterie pour n'en nommer que quelques-uns.

CASSARAM et CA4IOT (Context Awareness for Internet of Things) sont conçus pour sélectionner efficacement les capteurs [106]. L'interrogation sémantique et le raisonnement quantitatif sont utilisés pour effectuer des recherches et une sélection de capteurs sensibles au contexte. CASSARAM introduit également la comparaison des distances euclidiennes pondérées en fonction des priorités des utilisateurs dans un espace multidimensionnel afin que l'indexation et la comparaison soient plus rapides et plus efficaces. Cependant, les utilisateurs doivent toujours utiliser l'interface basée sur le curseur et la recherche SPARQL (c'est un langage sémantique de requête de bases de données pour récupérer et manipuler des données stockées dans le format Resource Description Framework (RDF)) pour d'autres données requêtes.

Martin Molina et coll. a proposé une nouvelle approche pour utiliser des données géographiques ouvertes pour générer une description en langage naturel pour les réseaux de capteurs hydrologiques [107] utilise la TLN et l'ontologie pour dériver des coordonnées géographiques (latitude et longitude) ainsi que le type de quantité physique mesurée par chaque capteur, ils utilisent l'ontologie pour générer un langage naturel à partir d'un raisonnement sémantique basé sur un modèle. Les informations de métadonnées de chaque capteur dans les réseaux de capteurs et la caractéristique géographique

5.3 Traitement du langage naturel avec l'ontologie

La façon la plus naturelle et intuitive de rechercher des informations est de poser des questions. Pour la plupart des gens, "*Quelle est la température moyenne à Casablanca du 2016 à 2019*" est assez claire. Comme mentionné par Kok-Kiong Yap et al, Lorsque les humains localisent des objets, ils ne le font pas en termes de coordonnées absolues, ils utilisent plutôt des repères identifiables [108]. Il y a bien sûr de nombreux défis ; cependant, la requête basée sur le langage naturel présente l'avantage d'être généralement spécifiée de manière vague.

Cela aide efficacement à cacher la nature hétérogène des réseaux de capteurs sous-jacents. Grâce au traitement, le système doit extraire suffisamment de contexte pour interpréter les plages de temps, sélectionner et fusionner correctement les données du capteur. Afin de gérer les réseaux de capteurs de différents domaines à plus grande échelle et de fournir une agrégation de niveau supérieur, nous définissons trois aspects d'une requête en langage naturel : la portée temporelle, spatiale et caractéristique des capteurs. (Ces aspects seront définis dans les sections suivantes). Un exemple de requête en langage naturel balisé incluant ces aspects est présenté dans Figure 5.1. Sur la base de ces trois étendues, nous avons conçu l'agent de requête TLN pour traiter les requêtes entrantes et générer des informations extraites pour l'agent suivant à l'aide du pipeline de traitement des requêtes.

Figure 5.1 : exemple des annotations de TALN

Les étapes de traitement TLN se compose de plusieurs annotateurs qui marquent le texte d'intérêt dans la requête. Chaque étendue a son propre annotateur pour traiter la requête à jetons. Le schéma de TLN est illustré à la figure 5.2.

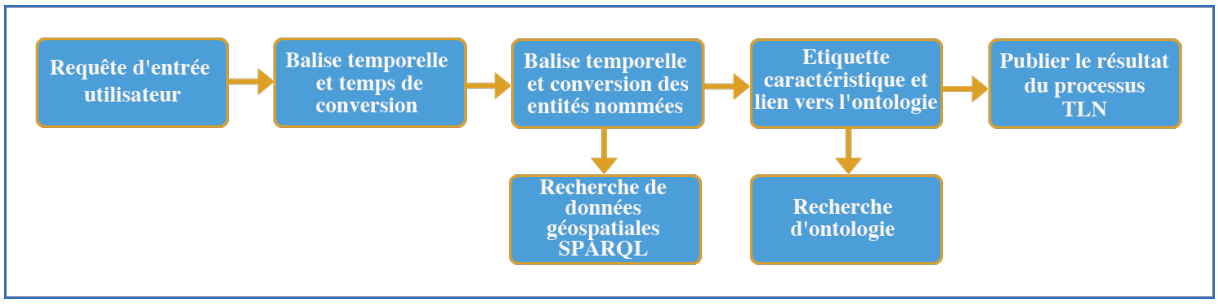

Figure 5.2 : Les étapes de traitement TLN pour extraire la portée temporelle, spatiale et caractéristique d'une requête en langage naturel.

Une fois que l'utilisateur a soumis la requête en langage naturel, l'annotateur temporel marquera d'abord les jetons liés au temps et les convertira en une plage d'horodatage ou une référence temporelle actuelle. Ensuite, l'annotateur spatial extraira les jetons spécifiant les noms d'emplacement ou d'organisation et générera une recherche SPARQL pour les convertir en valeurs telles que des coordonnées. Ensuite, l'annotateur caractéristique marquera les jetons liés au type de mesure du capteur ou à la fonction d'agrégation à l'aide d'une recherche d'ontologie. Enfin, le pipeline publiera le résultat dans la rubrique de requête désignée.

5.3.1 Portée temporelle

Comme la plupart des données de capteur sont basées sur des séries chronologiques, nous supposons pour l'instant que chaque ensemble de données de capteur possède des informations de référence temporelles explicites. Les données sans ces informations contiennent toujours des informations temporelles implicites définies sur l'ensemble de toutes les données. Ces informations implicites pourraient être utilisées comme référence temporelle pour la recherche. Par conséquent, la portée temporelle est essentielle à l'accès omniprésent aux données.

La portée temporelle décrit la plage de temps d'intérêt dans la requête en langage naturel. Nous définissons deux classes de requête temporelle: historique et streaming (diffusion). La portée historique est utilisée pour les requêtes entre deux instances de temps distinctes, survenues dans le passé, telles que: "*Quelle a été la pluviométrie totale de 2012 à 2014*". Cette requête sera convertie dans la plage 2012-01-01 au 2014-12-31. La portée de diffusion en continu est pour les conditions actuelles (en temps réel), telles que: "*Quelle est la température extérieure actuelle?*", Dans ce cas, la requête utilise le "streaming" pour capturer les données en temps réel. La demande de diffusion sera supprimée automatiquement lorsque la référence future temporelle expire.

Le raisonnement temporel relatif impose un autre défi à l'extraction d'informations temporelles car il peut représenter à la fois une plage de temps et une instance de temps. Par exemple, "*hier*" peut être interprété comme une plage horaire comprise entre 00:00:00 et 23:59:59 ou exactement 24 heures comme une instance de temps. Les étapes TLN interprètera les temps relatifs comme une instance de temps s'il y a une seconde instance de temps pour l'appairer afin de créer une plage. Sinon, il sera interprété comme la plage de temps appropriée.

5.3.2 Portée spatiale

La portée spatiale fait référence à l'emplacement physique des capteurs d'intérêt. Lors de la description des lieux, les gens ont tendance à utiliser des entités nommées et une plage relative plutôt que des latitudes et des longitudes numériques, comme "*à moins de 30 kilomètres de Casablanca*". La plupart des capteurs ne fournissent pas d'entités nommées dans leurs informations de métadonnées.

Au lieu de cela, ils ont généralement des coordonnées décrivant l'emplacement du capteur. Par conséquent, nous utilisons TLN pour extraire l'entité nommée (NE) de la requête, puis utilisons le géocodage pour déterminer les coordonnées à partir de l'entité nommée.

Pour permettre des requêtes spatiales plus générales telles que les noms d'organisations et les noms de bâtiments, nous avons besoin d'une base de données plus riche de noms et de coordonnées d'emplacement. L'ensemble de données de géocodage public contient une énorme quantité de données géographiques [109].

5.3.3 Portée caractéristique

Nous définissons le périmètre caractéristique comme toute information supplémentaire des métadonnées du capteur en plus des références temporelles et des emplacements. Il peut s'agir du type d'observation, de l'unité d'observation ou même de l'événement d'observation. Par exemple, la "vitesse moyenne du vent" et la "consommation d'énergie maximale" sont la portée caractéristique d'une requête de données de capteur en langage naturel.

Les ontologies utilisées dans SSN et SensorML ont une riche collection de types d'observations disponibles et d'autres informations de métadonnées. Par exemple, SWEET [110] de la NASA a plus de 2000 types d'observation.

Cependant, nous devons mapper les jetons de langage naturel aux types d'observation basés sur l'URL (Localisateur uniforme de ressource) car ces types d'observation sont partagés par de nombreuses ontologies différentes. Sur la base des noms définis dans l'ontologie, nous utilisons des modèles de jetons basés sur des règles pour mapper des mots clés de requête en langage naturel. Comme de nombreux types de mesure basés sur l'ontologie sont bien définis et proches du langage naturel, nous utilisons un modèle pour lier le jeton et l'URL. Par exemple, la "température de l'air" est mappé à "température de l'air" puis à http://mmisw.org/ont/cf/ parameter/air temperature.

Les fonctions d'agrégation, si elles sont spécifiées, font également partie de la caractéristique portée. Par exemple, pour satisfaire la requête en langage naturel "température maximale", nous récupérons d'abord les données brutes de tous les réseaux de capteurs appariés pour la "température", puis appliquons la fonction d'agrégation maximale.

5.4 Traitement du langage naturel (TLN)

Le traitement du langage naturel (TLN) est un domaine de l'informatique qui nous aide à déduire ce que l'utilisateur essaie de dire à travers ses commandes vocales. Le TLN de notre travail donne à l'utilisateur la liberté d'interagir avec les appareils électroménagers avec sa propre voix et son langage normal plutôt qu'avec des commandes informatiques compliquées. Le traitement automatique du langage naturel (TLN) est un domaine de l'informatique, de l'intelligence artificielle et de la linguistique informatique qui aide le système informatique à comprendre et à répondre aux commandes données en langage naturel (humain). La TLN peut être classée dans le domaine de l'interaction homme-machine. La plupart des défis de la TLN sont énumérés comme suit: compréhension du langage humain, permettra aux ordinateurs de tirer un sens des commandes vocales qui lui sont données par le biais du langage humain (naturel), et d'autres impliquent une interaction en langage naturel entre les ordinateurs et les humains. La plupart des derniers algorithmes TLN sont basés sur l'apprentissage automatique, en particulier l'apprentissage automatique statistique.

5.4.1 Arrêt de la suppression des mots

Le processus de conversion des données en quelque chose qu'un ordinateur peut comprendre est appelé prétraitement. L'une des principales formes de prétraitement consiste à filtrer les données inutiles. Dans le traitement du langage naturel, les mots (données) inutiles sont appelés mots vides [111]. Bien que les "mots vides" se réfèrent généralement aux mots les plus courants dans une langue, il n'y a pas de liste universelle unique de mots vides utilisés par tous les outils de traitement du langage naturel, et en effet, tous les outils n'utilisent même pas une telle liste.

5.5 Contexte linguistique

Définition : Le traitement automatique des langues naturelles (TALN) est un domaine de recherche interdisciplinaire qui vise à créer des programmes informatiques capables de traiter automatiquement des langues naturelles mettant en jeu du matériau linguistique. Cet objectif passe nécessairement par l'explicitation des règles de la langue puis leurs représentations dans un formalisme calculable et enfin leurs implémentations à l'aide des programmes informatiques.

5.5.1 Brève description de la langue amazighe :

La langue amazighe appartient à la famille des langues Afro-Asiatiques ou Chamito-Sémitiques (Chaker, 1991; Cohen, 2007). Elle est parlée au Maroc, en Algérie, Tunisie, Libye

et dans l'oasis égyptienne de Siwa. Elle est également parlée par beaucoup d'autres communautés dans certaines régions du Niger du Mali et du Burkina Faso et par les communautés amazighes immigrées partout dans le monde. Malgré sa longue histoire et son usage par une partie considérable de la population marocaine dans toutes leurs communications et affaires quotidiennes, l'amazighe est restée marginalisée pendant une longue durée. Cependant, avec l'émergence de la revendication identitaire, les locuteurs natifs militent pour la sauvegarde et la promotion de leur langue et de leur culture. Ainsi, le Roi Mohamed VI a pris l'initiative de créer un organisme académique à savoir l'Institut Royal pour la Culture AMazighe (IRCAM) chargé de la préservation et de la promotion de la culture amazighe. Cet institut a pour mission d'aménager la langue afin qu'elle puisse être enseignée et jouer un rôle véritable à l'échelle nationale.

5.5.2 Tifinaghe Unicode :

L'alphabet Tifinaghe-IRCAM comprend 33 graphèmes correspondant aux 33 phonèmes de l'amazighe standard et basé sur un système graphique à tendance phonologique. Ce système ne retient pas toutes les réalisations phonétiques produites, mais uniquement celles qui sont fonctionnelles [112]. Cet alphabet comporte :

- 27 consonnes dont : les labiales (ⵀ, ⵖ, ⴼ), les dentales (ⵜ, ⵟ, ⴷ, ⴹ, ⵍ, ⵔ, ⵚ, ⵣ), les Labiovélaires (ⵯ, ⵡ, ⵳, ⵷), les palatales (ⵛ, ⵊ), les vélaires (ⴽ, ⵅ), les labiovélaires (ⴽⵯ, ⵅⵯ), les uvulaires (ⵇ, ⵃ, ⵄ), les pharyngales (ⵄ, ⵃ) et la laryngale (ⴰ) ;
- 2 semi-consonnes : ⵢ et ⵡ;
- 4 voyelles : trois voyelles pleines ⴰ, ⵉ, ⵓ et la voyelle neutre (ou schwa) ⴻ qui a un statut assez particulier en phonologie amazighe.

Le sens de l'écriture est de gauche à droite. Les mots sont séparés à l'aide de l'espace typographique, les signes de ponctuation utilisés sont les mêmes signes conventionnels adoptés par les systèmes alphabétiques ayant la même orientation :

- Ponctuations : ({}), (<>), ("), (/), ..,
- Symbole typographiques : (@), (&), (*), (€), ..,etc.
- Symbole de numérations : (0), (1), (2), (3), (4), ..,etc.
- Symbole de Mathématiques : (≠), (√), (+), (÷), (‰),..,etc.

Les correspondances entre les différents systèmes d'écriture et les correspondances de translittération sont présentées dans le tableau 1.

Tifinaghe Unicode		Translitération		Caractères utilisés dans Tifinaghe IRCAM		Système d'écriture choisi
Code	Caractère	Latin	Arabe	Caractère	Codes	
U+2D30	ⴰ	a	ا	A, a	65, 97	a
U+2D31	ⴱ	b	ب	B, b	66, 98	b
U+2D33	ⴳ	g	گ	G, g	71, 103	g
U+2D33&U+2D6F	ⴳⵯ	g^w	گ+ُ [1]	Å, å	197, 229	g^w
U+2D37	ⴷ	d	د	D, d	68, 100	d
U+2D39	ⴹ	ḍ	ض	Ä, ä	196, 228	D
U+2D3B	ⴻ	e	يِ	E, e	69, 101	e
U+2D3C	ⴼ	f	ف	F, f	70, 102	f
U+2D3D	ⴽ	k	ک	K, k	75, 107	k
U+2D3D&+2D6F	ⴽⵯ	k^w	گ+ُ	Æ, æ	198, 230	k^w
U+2D40	ⵀ	h	ھ	H, h	72,104	h
U+2D43	ⵃ	ḥ	ح	P, p	80,112	H
U+2D44	ⵄ	E	ع	O, o	79, 111	E
U+2D45	ⵅ	x	خ	X, x	88, 120	x
U+2D47	ⵇ	q	ق	Q, q	81, 113	q
U+2D49	ⵉ	i	ي	I, i	73, 105	i
U+2D4A	ⵊ	j	ج	J, j	74, 106	j
U+2D4D	ⵍ	l	ل	L, l	76, 108	l
U+2D4E	ⵎ	m	م	M, m	77, 109	m
U+2D4F	ⵏ	n	ن	N, n	78, 110	n
U+2D53	ⵓ	u	و	W, w	87, 119	u
U+2D54	ⵔ	r	ر	R, r	82, 114	r
U+2D55	ⵕ	ṛ	ڕ	Ë, ë	203, 235	R
U+2D56	ⵖ	ɣ	غ	V, v	86, 118	G
U+2D59	ⵙ	s	س	S, s	83, 115	s
U+2D5A	ⵚ	ṣ	ص	Ã, ã	195, 227	S
U+2D5B	ⵛ	c	ش	C, c	67, 99	c
U+2D5C	ⵜ	t	ت	T, t	84, 116	t
U+2D5F	ⵟ	ṭ	ط	Ï, ï	207, 239	T
U+2D61	ⵡ	w	ۏ	W, w	87, 119	W
U+2D62	ⵢ	^+ي	ي	Y, y	89, 121	Y
U+2D63	ⵣ	z	ز	Z, z	90, 122	z
U+2D65	ⵥ	ẓ	ژ	Ç, ç	199, 231	Z
U+2D6F	ⵯ	w	ُ	Aucun correspondant		w

Tableau 5.1 : Système d'écriture choisi pour la translitération en latin

[1] Il représente le caractère arabe DAMMA reversé, dont le code est 0657.

5.6 Morphologie de la langue amazighe

La plupart des mots amazighes ont des racines consonantiques. Les racines des mots peuvent avoir une, deux, trois ou quatre consonnes ; parfois, ils s'étendent à cinq. Les mots sont créés à partir de la combinaison d'une racine et d'un schème. A partir d'une racine donnée, on peut avoir plusieurs dérivés verbaux et nominaux. La nature de la dérivation verbale peut être de plusieurs types: causatif, réciproque, réfléchi, passif, etc. Les dérivés nominaux sont: les noms d'agent, les noms d'action, les noms d'instrument, les noms de lieu, etc [113].

Les mots peuvent être classés en différentes classes grammaticales que nous citons: le nom, le verbe et les particules. Dans cet article, nous sommes intéressés par morphologie du nom.

Le nom amazigh est toujours composé d'un mot entre deux espaces et formé d'une racine et d'un motif. Il est caractérisé par le sexe (masculin ou féminin), le nombre (singulier ou pluriel) et l'état (libre ou constructif).

- Genre : Le nom amazigh est caractérisé par un genre grammatical: masculin ou féminin.
Exemple : awragh/jaune (masc), tawragh**t**/jaune(fém).

- Nombre : le nom, masculin ou féminin, a un singulier et un pluriel. Ce dernier a quatre

- Formes : le pluriel externe, le pluriel brisé, le pluriel mixte et le pluriel en ⵉⵏ [id].

- Le pluriel externe: est formé par une alternance de la première voyelle ⴰ / ⵉ [a / i] accompagné d'une suffixe de " ⵏ " [n] ou l'une de ses variantes.
Exemple : ⴰ*dlis « livre »* → ⵉdlisn " ***livres*** **".**

- Le pluriel brisé: implique un changement dans les voyelles du nom.
Exemple : ***adrar*** " ***montagne*** " → ***idurar*** " ***montagnes*** **".**

- Le mixte pluriel: est formé par le changement des voyelles accompagné, parfois par l'utilisation de la suffixe par ⵏ [n].
Exemple : ***uccn*** " ***chacal*** " → ***uccann*** "***chacals*** " ***/ urtu*** " ***verger*** " → ***urtan*** " ***vergers*** ".

- Le pluriel en ⵉⵏ [id]: ce type de pluriel est obtenu par préfixe ⵉⵏ [id]. Il est appliqué à un ensemble de noms comprenant: les noms avec une consonne initiale, les noms propres, le parent les noms, les noms composés, les chiffres, ainsi que les noms empruntés.
Exemple : ***murran*** " ***gaillard*** " → ***id murran*** " ***les gaillards*** **".**

- **Etat:** on distingue deux états: l'Etat libre et l'Etat constructif.

L'état libre: N'est pas marqué. Le nom est à l'état libre s'il est: un seul mot isolé de tout contexte syntaxique, un objet direct ou un complément de la particule prédictive ⴷ [d].
Exemple : ***d amhdar*** " ***c'est un étudiant*** " **/** ***d ta*** "***c'est celle-ci*** " ***/ d azugagh*** " ***c'est le rouge*** ".

L'état de construction: implique une variation de la voyelle initiale. Dans le cas des noms masculins, il prend l'une des formes suivantes: alternance de voyelle initiale ⴰ [a] / ⵓ [u] ou addition de ⵡ[w]; ajout de ⵢ [y] aux noms de voyelle ⵉ [i].
Exemple : ***argan*** "***arganiers***" → ***wargan*** "***arganiers***" ***/udmawn*** "***visages***" → ***wudmawn***.
Pour les noms féminins, il s'agit d'enlever la voyelle initiale ou de maintenir cette voyelle.
Exemple : ***ticirratin*** " ***filles***" →***tcirratin*** " ***filles***"

5.7 Défis et objectifs de la reconnaissance des nouvelles entités REN de l'amazighe

Nous utilisons une méthode hybride pour identifier les entités nommées dans les textes amazighs. L'autre contribution de ce travail est consacré à l'expérimentation de différentes caractéristiques morphologiques de l'amazighe pour l'apprentissage automatique ainsi que la combinaison de fonctionnalités.

Nous formulons quelques objectifs de ce travail:

1- Développer de nouvelles méthodes et fonctionnalités de reconnaissance pour améliorer Les performances de la langue amazighe.

2- Proposer une approche hybride pour améliorer l'adaptabilité du REN amazigh.

3- Expérimentez la désambiguïsation sur un petit sous-ensemble d'entités nommées sélectionnées.

4- Créer un système REN de qualité et réutilisable.

5- Adapter et appliquer cette approche pour les RCSFs intelligents

Ce travail se concentre principalement sur la langue amazighe et ses particularités du point de vue de la tâche REN.

Nous avons relevé de nombreux défis posés par les particularités de la langue amazighe, qui diffère considérablement des autres langues européennes. Nous passons en revue ci-dessous quelques problèmes qui doivent être pris en considération lors de la construction d'un système NER pour les Amazighs.

- **Ambiguïté :** De nombreux mots peuvent être interprétés de plusieurs manières, produisant des significations différentes. Afin d'atténuer l'impact de ce problème, des informations contextuelles seront utilisées dans notre système.
- **Absence de majuscules :** Contrairement aux langages de script latins, l'amazighe ne fait pas de distinction entre les lettres majuscules et minuscules (les majuscules identifient le début et la fin des NE potentiels dans la plupart des langages des scripts latins).
- **Morphologie complexe :** La langue amazighe a une structure morphologique très systématique mais complexe basée sur des schémas racinaires et est considérée comme une langue hautement flexionnelle. Habituellement, un lemme donné en amazigh peut avoir plus d'une forme de mot qui comprend une racine, des préfixes, des suffixes. Ce problème doit être traité afin de détecter correctement les éléments EN dans le texte.
- **Manque de standardisation de l'orthographe amazighe :** Le texte amazighe, comme de nombreuses autres langues, a de nombreuses variantes orthographiques lorsqu'il s'agit des noms et surtout les noms étrangers, qui ne contiennent pas une orthographe standardisée.
- **Manque de ressources linguistiques:** nous menons une étude sur les ressources en langue amazighe et les outils de TALN (par exemple, corpus, répertoires géographiques, étiqueteurs de point de vente, etc.).
  Cela nous a amenés à conclure qu'il existe une limitation du nombre de ressources linguistiques amazighes disponibles par rapport aux autres langues.

Bon nombre de ceux disponibles ne sont pas pertinents pour les tâches REN amazighe en raison de l'absence d'annotations EN dans la collecte de données. Les répertoires géographiques amazighs sont également rares et de nombre limité.

Par conséquent, nous avons tendance à construire nos ressources linguistiques amazighes afin de former et d'évaluer cette langue.

5.8 Corpus amazighe de test

Tout d'abord, la construction d'un système d'extraction d'entités nommées nécessite de collecter un nombre suffisant de textes qui serviront non seulement de corpus de formation (pour établir les règles), mais également de corpus de test. Comme nous l'avons mentionné précédemment, il n'y a pas de corpus amazigh disponible pour la tâche REN. Pour cette raison, nous avons construit notre propre corpus. Il contient les Actualités régionales (11 articles), Economie (27 articles), Social (31), Actualités politiques (25), Sport (33), activités mondiales (23 articles) et quelques actualités générales (36 articles). Ainsi, nous avons collecté 402 articles de ces catégories au format html et nous les avons tous concaténés dans un seul fichier texte. Il contient 78 220 jetons.

L'annotation morphosyntaxique d'un corpus nécessite l'adoption d'un jeu des étiquettes (Tagest), pour le cas de la langue amazighe le jeu des étiquettes utilisé est illustré dans le tableau 5.2 :

N°	TAG	Désignation
1	NN	Nom commun
2	NNK	Nom de parenté
3	NNP	Nom propre
4	VB	Verbe sous la forme de base
5	VBP	Verbe sous la forme participe
6	ADJ	Adjectif
7	ADV	Adverbe
8	C	Conjonction
9	DT	Déterminant
10	FOC	Focaliseur
11	IN	Interjection
12	NEG	Particule négative
13	VOC	Vocatif
14	PRED	Particule de prédiction
15	PROD	Particule d'orientation
16	PRPR	Particule préverbale
17	PROT	Autre particule
18	PDEM	Pronom démonstratif
19	PP	Pronom personnel
20	PPOS	Pronom possessif
21	INT	Interrogatif
22	REL	Relatif
23	S	Préposition
24	FW	Mot étranger
25	NUM	Numérique

26	DATE	Date
27	ROT	Autre résiduel
28	PUNC	Ponctuation

Tableau 5.2 : Jeux des étiquettes grammaticales détaillées de la langue amazighe

5.9 Notre approche NER Amazighe

Les recherches récentes en (REN) ont tendance à utiliser des approches d'apprentissage automatique. Les méthodes d'apprentissage comprennent divers apprentissages supervisés, semi-supervisés et non supervisés. L'apprentissage supervisé a tendance à être la technique dominante pour la reconnaissance et la classification des entités nommées. Cependant, les méthodes d'apprentissage automatique supervisé nécessitent une grande quantité de documents annotés pour la formation de modèles et leurs performances dépendent généralement de la disponibilité de données de formation de haute qualité suffisantes dans le domaine d'intérêt. Il existe certains systèmes qui utilisent des méthodes hybrides pour combiner différents systèmes basés sur des règles et / ou d'apprentissage automatique pour améliorer les performances par rapport aux approches individuelles. Les systèmes hybrides utilisent au mieux les bonnes caractéristiques des différents systèmes ou méthodes pour obtenir les meilleures performances globales.

5.9.1 Système NER typique

Un identificateur d'entité nommé typique comporte quatre éléments principaux, qu'il soit conçu selon une approche basée sur des règles ou une approche d'apprentissage automatique. L'architecture d'un système REN typique est illustrée à la figure 5.3. Les principaux éléments sont la segmentation ou bien la tokenisation, le traitement morphologique et lexical, l'identification et la classification.

La segmentation est la première étape de l'interprétation du texte en fractionnant une chaîne de mots / caractères (comprenant un document, paragraphe ou une phrase) en parties minimales de texte structuré qui sont utiles pour être utilisées comme une unité, appelée jeton.

ar	a	-	-	-	r	-	-	-	PRPR
i	-	-	-	-	-	-	-	-	S
s	-	-	-	-	-	-	-	-	PP
ttHyyaln	t	tt	ttH	ttHy	n	ln	aln	yaln	VB
i	-	-	-	-	-	-	-	-	S
tmGra	t	tm	tmG	tmGr	a	ra	Gra	mGra	NN
ann	a	an	-	-	n	nn	-	-	DT
sg	s	-	-	-	g	-	-	-	S
usggwas	u	us	usg	usgg	s	as	was	gwas	NN

lli	l	ll	-	-	i	li	-	-	REL
izrin	i	iz	izr	izri	n	in	rin	zrin	VBP
.	-	-	-	-	-	-	-	-	PUNC

Tableau 5.3 : Extrait du corpus annoté suivant un jeu d'étiquettes

Une fois le processus de segmentation est terminé, le traitement morphologique et lexical se poursuit, Il utilise principalement un étiqueteur de partie du discours et chaque mot d'une séquence de mots est étiqueté avec une étiquette intérieure ou extérieure. En outre, il utilise des composants tels que le découpage EN et l'extraction de fonctionnalités. Le traitement morphologique et lexical aide principalement à la détection des EN qui sont représentée sur la figure 5.3 comme identification. Le composant d'identification détecte les éléments NE à l'aide de modèles ou de règles stockés, en fonction de l'approche utilisée à savoir le SVM (Support Vector Machines) et CRF (Conditional Random Fields).

Les éléments NE détectés sont alors prêts à être classés dans leurs classes respectives. L'étape de classification prend les éléments EN détectés et les classe dans leurs catégories correspondantes. La classification est effectuée par un classificateur.

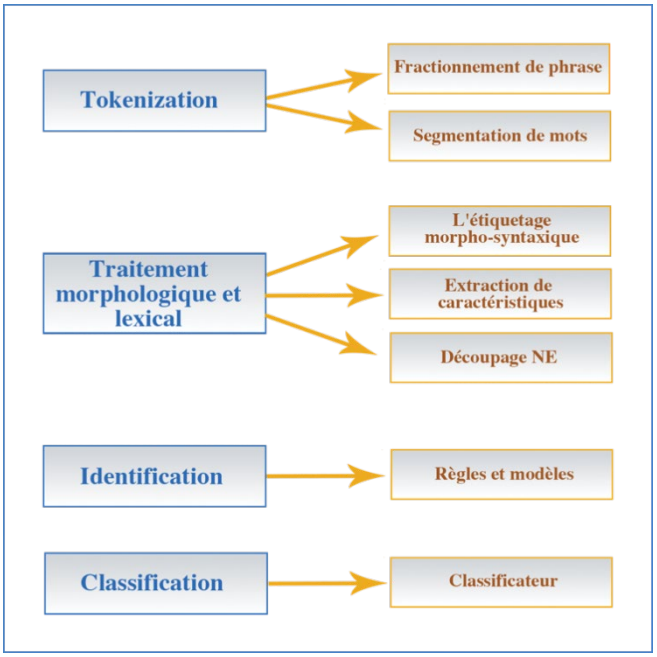

Figure 5.3 : architecturetypique d'un système REN

5.9.2 Notre approche

Notre travail a suivi une approche hybride avec le composant d'apprentissage automatique basé sur un algorithme CRF (Conditional Random Fields). L'architecture du système comporte deux processus principaux (figure 5.4): les processus d'apprentissage et de prédiction.

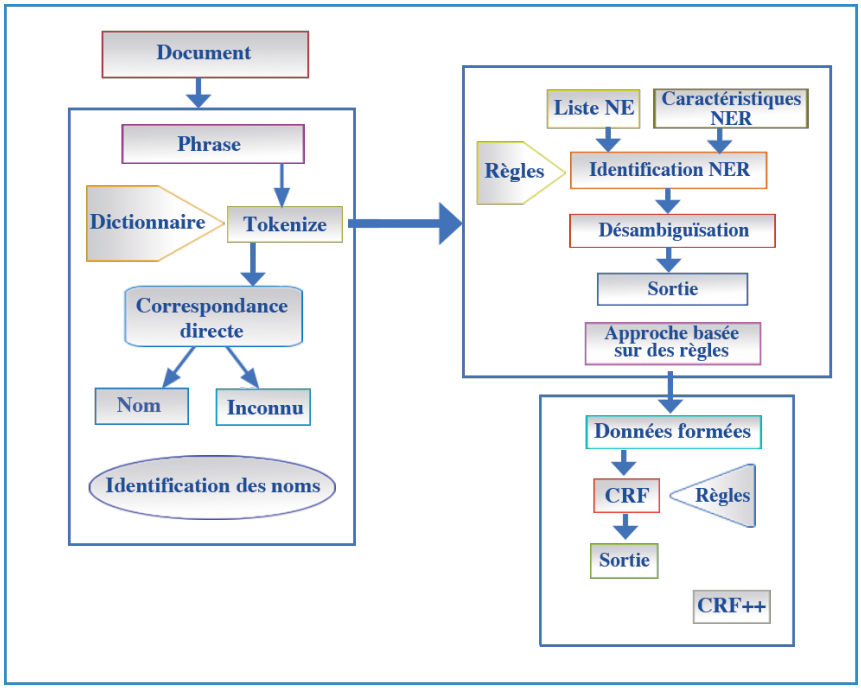

Figure 5.4 : Architecture de REN

- Le processus d'apprentissage fonctionne sur les données de formation et l'utilisé pour générer le modèle formé.

- Le processus de prédiction est un processus qui fonctionne sur le texte d'entrée fourni par un utilisateur et qui vise à reconnaître les éléments EN du texte.

Le système comprend également trois phases principales:

- La phase de prétraitement est celle où les données de formation et le texte brut sont prétraités pour la tâche suivante : La tokenisation et la segmentation sont les principales tâches de prétraitement, la première s'applique au corpus et la seconde au texte d'entrée simple.
- La phase de formation comprend des composants essentiels qui sont utilisés pour générer une séquence de jetons / tags, extraire des fonctionnalités, fragmenter les jetons et estimer le modèle avec les données de formation.
- La phase de reconnaissance est celle où le texte d'entrée prétraité comme une entrée, et le texte reconnu EN est une sortie, et il se compose de composants tels que le modèle formé, les règles stockées et les EN détectées.

Algorithme NER :

Étape 1: *Divisez le fichier d'entrée en phrases ;*

Étape 2: *Tokenisation* ;

Étape 3: *Si les jetons correspondent directement au dictionnaire, attribuez-les en tant que nom ;*

Étape 4: *Si le nom correspond à la liste REN, attribuez sa balise, sinon utilisez les fonctionnalités REN et les règles de désambiguïsation ;*

Étape 5: *Ayez toujours l'ambiguïté et les mots inconnus, continu.*

5.9.3 Champs aléatoires conditionnels (CRFs)

Les CRFs sont des modèles graphiques probabilistes, se basant sur la théorie de graphes et la théorie des probabilités. Ces deux théories permettent de modéliser le problème de classification des séquences : la théorie des graphes permet la modélisation des structures de séquence des étiquettes des phrases ; la théorie des probabilités, elle, permet de générer les ambiguïtés causées par les séquences des étiquettes. La modélisation des CRFs est faite sous forme de graphe permettant de contextualiser les relations entre les étiquettes. La tache consiste à trouver la relation la plus probable du champ aléatoire correspondant à l'étiquetage.

Les CRFs ont été introduits dans [114]. L'idée de CRF est fortement basée sur ME (Markov à Entropie).

La différence est que ME classe une instance après l'autre tandis que CRF classe la séquence entière à la fois.

Écrit mathématiquement, ME estime la probabilité conditionnelle $p(y_i / x_i)$ pour $i = 1,..,n$ et CRF estime $p(y / x)$ où y et x sont des vecteurs à n dimensions. La probabilité $p(y / x)$ peut être calculée en utilisant des matrices et une variante de l'algorithme avant-arrière.

Les fonctionnalités sont étendues et peuvent utiliser l'état précédent contrairement à ME. Deux types de fonctions sont utilisés, les états s et la transition t. Les entités d'état peuvent être considérées comme un sous-ensemble d'entités de transition, où l'état précédent n'est pas utilisé, et une définition générale des fonctionnalités peut être utilisée. Des tests initiaux sur la tâche REN ont été effectués en [115].

Depuis leur introduction, de nombreux systèmes ont utilisés avec de très bons résultats [116, 117]. Le CRF est considéré comme la méthode de classification la plus efficace pour le REN. Depuis leur introduction, de nombreux systèmes les ont utilisés avec de très bons résultats [18, 19]. Le CRF est considéré comme la méthode de classification la plus efficace pour le REN.

5.9.4 Algorithmes utilisés

5.9.4.1 Algorithme pour l'identification des noms

Notre algorithme pour l'identification des noms est décrit ci-dessous. Nous supposons que nous avons une petite quantité de données étiquetées et un classificateur qui est formé sur ces données amazighes. Nous exploitons un grand corpus non étiqueté à partir du domaine de test à partir duquel nous ajoutons automatiquement et progressivement de nouvelles données de formation, de sorte que notre corpus possède deux propriétés:

i**)** Étiqueté avec précision, ce qui signifie que les étiquettes attribuées par annotation automatique des données sélectionnées non étiquetées sont correctes.

ii**)** Non redondant, ce qui signifie que le classificateur devrait s'améliorer de façon monotone les données d'entraînement et quelles sont mises à jour.

Algorithme pour l'identification des noms :

Étape 1: *Lire le fichier d'entrée et décomposer-le en phrases ;*

Étape 2: *Lisez chaque phrase et divisez-la en jetons ;*

Étape3: *Lire chaque jeton ;*

Étape 4: *Pour chaque vérification de boucle (jeton) avec le dictionnaire amazigh ;*

Étape 5: *S'il correspond directe avec le dictionnaire, attribuez un nom ;*

Étape 6: *Sinon si aucune correspondance avec le dictionnaire, vérifiez avec la liste des*

Suffixes de noms ;

Étape 7: *Si des suffixes sont trouvés et que la racine se trouve dans le dictionnaire amazigh, attribuez-lui un nom ;*

Étape 8: *Sinon, si le suffixe correspond et que la racine n'est pas trouvée, le jeton peut être un nom ;*

Étape 9: *Sinon attribuer la catégorie "inconnu" Fin de boucle.*

5.9.4.2 Algorithme d'identification REN

Étant donné que les fonctionnalités de chaque jeton incluent les fonctionnalités copiées à partir de ses voisins, en plus de celles extraites du jeton lui-même, ses voisins doivent également être ajoutés à l'ensemble d'apprentissage. Si la confiance des voisins est faible, les voisins seront supprimés des données de formation après avoir copié leurs caractéristiques sur le jeton d'intérêt. Si les scores de confiance des voisins sont élevés, nous nous étendons davantage aux voisins des voisins jusqu'à ce que les jetons de faible confiance soient atteints. Nous supprimons les voisins à faible confiance afin de réduire les chances d'ajouter des exemples de formation avec de fausses étiquettes.

Algorithme d'identification REN :

Étape 1: *Lire la liste des noms identifiés;*

Étape 2: *Vérifiez les listes du répertoire géographique pour les fonctionnalités REN;*

Étape 3: *Pour chaque boucle (nom) Si des caractéristiques de suffixe sont trouvées, attribuez une étiquette REN;*

Étape 4: *Sinon si des fonctionnalités de préfixe ont été trouvées, attribuez une balise REN;*

Étape 5: *Sinon si des fonctionnalités de contexte sont trouvées, attribuez une balise REN;*

Étape 6: *Sinon, s'il est trouvé dans la liste NER, attribuez une étiquette REN;*

Étape 7: *Sinon attribuez "Mot divers";*

Étape 8: *Si une ambiguïté est trouvée, alors appelez les règles de levée d'ambiguïté;*

Étape 9: *Sinon, si ambiguïté et mots inconnus trouvés puis appeler la boucle de fin CRF.*

5.10 Ressources linguistiques pour NER amazighe

5.10.1 Création de répertoires géographiques

Notre système Amazigh REN rassemble quatre répertoires géographiques différents fabriqués manuellement:

- Répertoire géographique des personnes: nous avons construit une liste d'environ 1120 entrées de noms amazighs et de noms étrangers transcrits en amazigh, extrait de notre corpus et de nos ressources Internet.

- Nomenclature géographique: nous considérons le type " lieu" ou nom de lieu comme: pays, villes, rivières, montagnes, océans et mers. Ainsi, nous avons développé un lexique contenant 2083 entrées, trouvées sur Internet, et extrait de notre corpus.

- Répertoire géographique des organisations: le lexique des organisations se limite à une liste de 330 noms d'entreprises et d'organisations que nous avons extraits du Web et de notre corpus.

- Divers (MISC): nous avons intégré dans cette classe jour et mois (cela contient 19 entrées), des nombres transcrits en langue amazighe (cela contient 87 entrées).

5.10.2 Mots déclencheurs

Les mots déclencheurs sont des mots qui ne sont pas des éléments EN, mais se trouvent souvent à proximité des éléments EN. Par exemple, "rrays" (président en français) peut être un mot déclencheur pour la personne EN.

La liste des mots déclencheurs peut être apprise automatiquement à partir des corpus ou peut être faite à la main.

Déclenchement des propriétés des composants du EN en cours de classification (par exemple, pour l'entité "banka n lmaghrib" (Banque du Maroc), nous pourrions avoir une fonctionnalité EN: trig = ORG). Modèles de contexte à gauche du EN, où chaque mot est marqué avec ses propriétés de déclenchement, ou avec une étiquette de mot fonctionnel si approprié (par exemple, la phrase (rrays n marikan) "Le président des États-Unis" produirait le modèle f ORG f pour le EN "États-Unis", en supposant que le mot "rrays" (président) est répertorié comme un déclencheur possible pour ORG).

5.11 Résultats et discussion

Le corpus a été divisé en 90% pour l'ensemble d'apprentissage et l'ensemble restant est destiné aux tests où l'ensemble d'apprentissage représente les valeurs d'entrée pour le modèle de classification de CRF. De plus, le corpus représente les entrées de données dans ce modèle. Le but des expériences présentées est d'évaluer les performances de notre approche hybride.

Le fichier du corpus total est divisé en quatre fichiers parmi lesquels trois doivent être utilisés conformément aux règles et les règlements de la tâche partagée dans CoNLL 2002 [118].

Les quatre fichiers sont nommés fichier de formation, fichier de développement, fichier de test et fichier d'expérimentation.

Les méthodes d'apprentissage sont formées avec les données de formation. Les données du fichier de développement sont utilisées pour régler les paramètres des méthodes d'apprentissage. Lorsque les meilleurs paramètres sont trouvés, la méthode peut être entraînée sur la formation des données et testé sur les données de test. Dans ce cas, la répartition entre les données de développement et de test ont été choisies pour éviter que les systèmes soient réglés sur les données de test. Le fichier d'expérimentation est utilisé plus tard dans l'expérimentation. Les statistiques du corpus amazigh EN développé sont présentées dans le tableau 5.5.

	Total Size	**Total NE**	**PER**	**LOC**	**ORG**	**DIVERS**
Fichier de Formation	8000	1206	411	266	296	233
Fichier de développement	5000	745	167	248	183	147
Fichier de teste	4000	704	234	235	91	144
Fichier d'expérimentation	6100	920	252	346	221	101

Tableau 5.5: Statistiques du corpus amazigh EN

Les systèmes REN sont généralement évalués à l'aide de trois paramètres d'évaluation: précision, rappel et F-mesure.

Tous sont représentés sous forme de pourcentage:

- Précision (*P*) calcule le pourcentage d'EN correctement reconnus sur le total des EN reconnus.
- Rappel (*R*) calcule le pourcentage des EN reconnus à partir de l'ensemble de référence.

Trois valeurs peuvent nous aider à calculer facilement les métriques d'évaluation de notre système (voir les équations ci-dessous).

Ce sont les vrais positifs (t_p), les faux positifs (f_p) et les faux négatifs (f_n)

$$Précision = \frac{t_p}{t_p + f_p}$$

$$Rappel = \frac{t_p}{t_p + f_n}$$

$$F\text{-}mesure = 2\frac{Précision.Rappel}{Précision + Rappel}$$

• t_p compte le nombre d'EN reconnus par un système REN et trouvés dans les données de test.

• f_p compte le nombre d'EN qui sont mal reconnus par un système REN mais qui ne sont pas dans le test.

• f_n compte le nombre d'EN qui ne sont pas reconnus par un système REN mais qui figurent dans les données de test.

Le système REN développé a été conçu selon la définition de l'ensemble d'étiquettes de tâches partagées CoNLL2002 et CoNLL2003 est formé par des étiquettes appartenant aux quatre catégories suivantes :

1- Noms des personnes: Notre corpus annoté contenait 1120 occurrences de noms de personnes. Parmi eux, 830 noms de personnes ont été correctement annotés, 30 étaient partiellement corrects. 231 noms de personnes n'ont pas été annotés et 29 faux positifs.

Les erreurs étaient en grande partie dues à:

i) Prénom et nom de la personne apparaissant dans le corpus, mais non inclus dans les listes de mots.

ii) Certains termes en langue amazighe sont des noms de personnes qui peuvent également être des noms de villes.

2- Emplacements: Il y avait 2042 noms d'emplacements dans le corpus. La méthode d'annotation a correctement identifié 1905 d'entre eux. Cependant, 51 noms de lieux ont été manqués, 56 des annotations n'étaient que partiellement correctes et il y avait 30 faux positifs. Le manque des normes d'écriture des noms d'emplacement implique des difficultés à reconnaître les entités nommées d'emplacement.

3- Organisations: Il y avait 622 noms d'organisations dans le corpus. 295 noms d'organisations ont été correctement annotés, 272 étaient partiellement corrects, 53 noms d'organisations n'ont pas été reconnus et il y a eu 5 faux positifs.
Cela est principalement dû au fait que nous utilisons la délimitation des éléments EN en utilisant uniquement des informations contextuelles qui ne sont pas suffisantes dans la tâche NE.

4- Nom Divers: désigne les éléments EN divers qui n'appartiennent à aucune des classes précédentes et incluent la date, l'heure, le nombre, les expressions monétaires, les expressions de mesure et les pourcentages.

Il y a 828 expressions de date / nombre dans le corpus. Parmi ceux-ci, 320 ont été correctement annotés, 470 étaient partiellement corrects et 38 ont été manqués. Il y a eu 0 faux positifs.

En utilisant les trois mesures, les résultats de notre reconnaissance d'entité nommée amazighe pour chaque type d'entité sont présentés dans le tableau 5.4 et la figure 5.5.

Entité nommée (EN)	Précision (%)	Rappel(%)	F-mesure(%)
Personnes	93	96	83
Emplacements	97	97	97
Organisations	74	77	76
Nom Divers	65	68	67

Tableau 5.4. Performances du système

Les résultats de précision ont été très satisfaisants: la précision d'extraction pour la catégorie Personne était de 93%, pour la catégorie Organisation 74%, pour la catégorie Emplacements 97% et pour la catégorie Divers 65%.

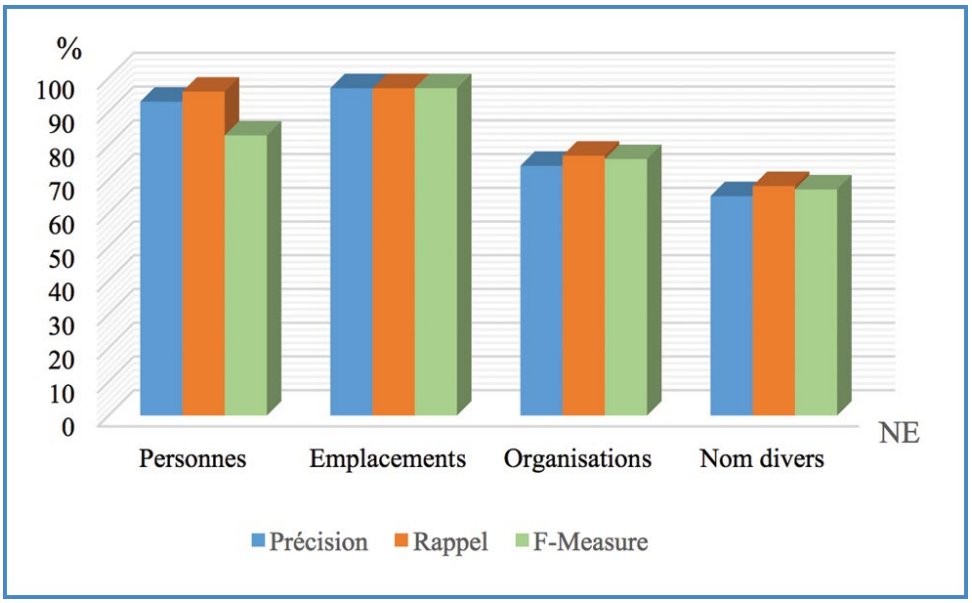

Fig 5.5 Performance Amazigh NER

L'objectif principal de notre travail est de construire des extracteurs d'entités amazighes de très haute précision pour les catégories Personne, Lieu, Organisation et Divers qui minimiseraient la sortie bruyante (entités et leurs relations). Nous avons utilisé les quatre catégories spécifiques, elles sont parmi les catégories les plus utilisées dans les systèmes de recherche d'informations à travers les domaines, elles peuvent donc être utilisées pour améliorer encore la précision du système REN et étendre sa portée grâce à l'apprentissage automatique.

La qualité de notre système d'extraction dépend cependant de la qualité des listes EN. Si les catégories n'ont pas un nombre fini de "Membres", notre méthode ne permettrait pas d'obtenir des résultats similaires de haute précision. La création de telles listes nécessite des recherches et du temps et peut varier d'une langue à l'autre. Cette méthode pourrait représenter un défi pour les très gros systèmes à forte intensité de données. Il ne serait cependant pas très difficile de prendre une liste amazighe d'EN et de trouver la liste équivalente dans d'autres langues. Notre

approche est suffisamment simple pour pouvoir être utilisée dans différentes langues autres que la langue amazighe. Notre méthode peut être appliquée dans tous les domaines d'application notamment les réseaux de capteurs sans fils au du routage.

5.12 Conclusion et perspectives

Le système REN pour la langue amazighe est difficile en raison de divers problèmes comme la nature inhérente, les ambiguïtés dans les classes d'entités nommées, les dépendances non locales, les apparitions de mots étrangers, les variations d'orthographe, etc. Ce chapitre a exploré diverses méthodologies et techniques, qui peuvent être utilisées dans la conception d'une entité nommée amazigh système de reconnaissance.

Le travail a présenté une tentative de développer le modèle de reconnaissance d'entité nommée pour la langue amazighe en utilisant une technique hybride. Le but de ce modèle est d'améliorer la précision du NER en langue amazighe introduit par différentes approches dans la littérature. Le corpus a été divisé en 90% pour l'ensemble d'apprentissage et l'ensemble restant est utilisé pour tester où l'ensemble d'apprentissage représente les valeurs d'entrée pour le modèle de classification de CRF. Le résultat a montré que notre approche surmonte d'autres méthodes dans leurs performances et en termes de précision. L'approche hybride atteint 93%, 97%, 74% et 65% pour la précision en Personne, Lieu, Organisation et Divers respectivement.

Le développement futur impliquera l'ajout de règles de grammaire afin d'obtenir un score plus élevé. Un autre domaine qui nécessite des recherches supplémentaires est celui des méthodes non supervisées et semi-supervisées pour le REN amazigh.

Autres objectifs en perspective de ce chapitre est de construire des extracteurs d'entités amazighes de très haute précision pour les noms pour différentes catégories, et leurs dépoilements dans le langage des réseaux de capteurs sans fils, on essayons d'utiliser une approche d'agrégation simple pour répondre à des questions telles que : "quel bâtiment a consommé le plus d'énergie hier ? " et "où est l'endroit le plus froid ? ", dans cette perspective, nous effectuons des recherches sur les algorithmes d'agrégations des capteurs pour choisir le plus adapté à notre approche hybride, en suite nous sélections un système adéquat pour interroger les données du RCSFs. (Comme le langage de requête SPARQ par exemple).

Dans l'avenir, nous continuerons d'étendre nos expériences au profil de la langue amazighe, non seulement en termes de production de données et de comparaisons, mais également en termes de conception et de mise en œuvre d'architectures d'apprentissage automatique, qui sont plus désireuses d'extraire des relations significatives sur lesquelles repose un mot extrait. Cette approche s'appuiera sur les avancées récentes et les enseignements tirés des sciences cognitives et de l'apprentissage robotique de type humain [119], où un robot apprend des éléments de sa mémoire sémantique et épisodique par l'interaction du langage avec les gens. Cet apprentissage de type humain peut se produire lorsque nous extrayons, représentons et raisonnons sur la signification des énoncés de langage naturel de l'utilisateur.

On conclut que les TICs (Technologies d'Informations et de Communications) constituent une piste prometteuse pour lutter contre la fracture numérique dont l'amazighe à souffert.

Chapitre

6

Méthode améliorée de cartographie probabiliste de Koblitz dans le cryptosystème à courbe elliptique : étude comparative et résultats.

6.1 Introduction :

Les courbes elliptiques (EC) ont été étudiées en mathématiques depuis longtemps, mais leur utilisation dans les applications cryptographiques a été suggérée pour la première fois par Neal Koblitz [81] et Victor Miller [82] qui ont confirmé les propriétés avantageuses du système EC qui conduit à des méthodes d'implémentation efficaces.

Avant que ECC ne devienne populaire, presque tous les algorithmes à clé publique étaient basés sur RSA, DSA (Digital Signature Algorithm) et DH (Deffie-Hellman) qui sont des cryptosystèmes alternatifs basés sur l'arithmétique modulaire. Ces cryptosystèmes sont toujours très importants aujourd'hui et sont souvent utilisés aux côtés de l'ECC. Alors que RSA et les autres cryptosystèmes peuvent être expliqués par les implémentations approximatives qui peuvent être écrites assez facilement, par contre les règles de base de la cryptographie utilisant ECC restent mystérieuses et incompréhensibles pour la plupart.

La sécurité de ECC, basé sur la difficulté de résoudre le problème du logarithme discret à courbe elliptique (ECDLP), offre une force de protection similaire en comparaison à d'autres cryptosystèmes à clé publique, sauf qu'elle nécessite une taille de clé considérablement plus petite.

Le codage (conversion d'un message en texte brut en coordonnées des points CE prédéfinis) et le décodage (conversion des coordonnées des points CE prédéfinisde en message en texte brut) sont des fonctions importantes dans les schémas de cryptage et de décryptage utilisant ECC avant de l'envoyer dans les canaux non sécurisés.

Ce chapitre fournit les connaissances et les conventions de base pour comprendre ce qu'est la cryptographie à courbe elliptique.

L'objectif principal de notre contribution est de présenter une amélioration de la méthode de cartographie probabiliste de Koblitz en utilisant un exemple de message écrit en caractères Tifinagh Unicode. L'amélioration proposée porte sur les variables de codage de l'algorithme de Koblitz pour faciliter la recherche des racines carrées de l'équation CE correspondant aux blocs du message M. Pour confirmer l'efficacité de l'algorithme proposé, nous l'avons comparé à deux méthodes. La première est basée sur un point générateur G d'un sous-groupe de CE avec une matrice pour changer la position des points; et la seconde utilise un schéma de code ECC hybride codé par l'ADN. Les résultats expérimentaux de la comparaison ont montré que notre algorithme de cartographie Koblitz modifié nécessite beaucoup moins de temps de codage qui ne varie pas nécessairement avec la taille du message M. Ces résultats peuvent augmenter la vitesse de mappage ce qui réduit la consommation d'énergie pour les cryptosystèmes basés sur CE.

Ce chapitre est ordonné comme suit: La section 2 fournit une définition claire des courbes elliptiques et son arithmétique sur les nombres réels. Dans les sections 3, 4 et 5 nous définissons : les points CE sur un champ fini, un groupe, l'ordre du sous-groupe, le générateur d'un groupe de courbe elliptique, et la sélection d'un point générateur approprié pour générer un sous-groupe cyclique avec un algorithme de multiplication optimisé. Dans la section 6 et 7 nous décrivons le système cryptographique basé sur une courbe elliptique utilisant également l'approche proposée de codage et de décodage (cartographie de Koblitz modifié).
La section 8 présente une brève définition de la langue amazighe et son système de codage et d'écriture au Maroc. La section 9 fournit la mise en œuvre de la méthode proposée (Koblitz). La section 10 décrit la méthode de cartographie (mapping) de la référence [90]. Les résultats de comparaison, les discussions, les propriétés et l'analyse sont présentés dans les sections 11 ,12 et 13. Enfin une conclusion est donnée dans la section 11.

6.2 Définition de la courbe elliptique CE

Une courbe elliptique peut être définie comme une courbe projective lisse de degré 3 dans le plan projectif, munie d'un point origine O ; l'ensemble des points muni d'une structure de groupe. La description la plus concrète provient du fait qu'on peut écrire leur équation affine sous la forme :

$$y^2 = x^3 + ax + b \quad \text{où} \quad 4a^3 + 27b^2 \neq 0 \qquad (1)$$

Il existe également des courbes elliptiques définies sur d'autres domaines pour la cryptographie, mais pour notre objectif et en vue d'exclure les courbes singulières, une courbe elliptique sera simplement l'ensemble des points décrits par l'équation : (1)

L'équation (1) est appelée la forme normale de Weierstrass pour les courbes elliptiques.

Par exemple la figure 1 regroupe 3 courbes différentes :

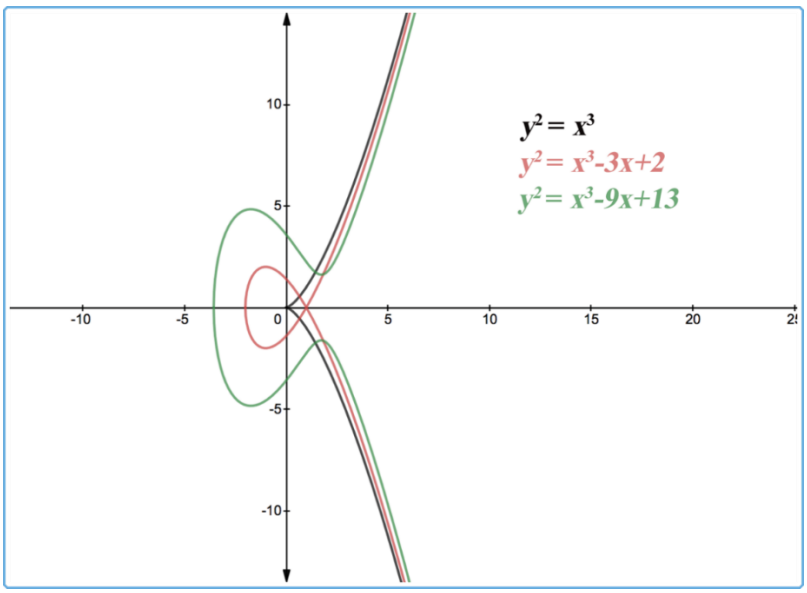

Figure 6.1 : Courbes elliptiques pour différentes valeurs des paramètres a et b.

- Une courbe avec une cuspide ($y^2 = x^3$),
- Une courbe avec une auto-intersection ($y^2 = x^3 - 3x + 2$),

Aucune d'entre elles n'est une courbe elliptique valide (ces deux courbes sont singulières).

La courbe verte ($y^2 = x^3 – 9x + 13$) parmi les formes normales de Weierstrass pour les courbes elliptiques. Selon la valeur de a et b, les courbes elliptiques peuvent avoir différentes formes sur le plan (x, y). Il est c'est facile de vérifier que les courbes elliptiques sont symétriques par rapport à l'axe des x.

- Une caractéristique étonnante des courbes elliptiques réside dans l'existence d'un moyen naturel de prendre deux points sur une courbe et de les "additionner" pour donner un troisième point [83]. Le mot additionner n'a rien avoir avec l'addition habituelle (+), mais il s'agit d'une opération qui combine deux points, d'une manière analogue à l'addition et qui a les propriétés connues pour la loi (+) à savoir la commutativité, l'associativité et l'existence d'un élément neutre. La manière la plus normale de décrire la "loi additionnelle" sur les courbes elliptiques consiste à utiliser la géométrie.

- Soit P et Q deux points sur une courbe elliptique $E(a, b)$, comme l'illustre la figure 6.2, commençons à dessiner la droite (L) qui coupe la courbe E en trois points, P, Q et R. Prenons la projection du point R par rapport à l'axe des abscisses, pour obtenir un nouveau point R' qui s'appelle la "somme de P et Q".

  Pour l'instant, désignons cette étrange loi d'addition par le symbole $\oplus$. Ainsi, nous écrivons : $R' = P \oplus Q$.

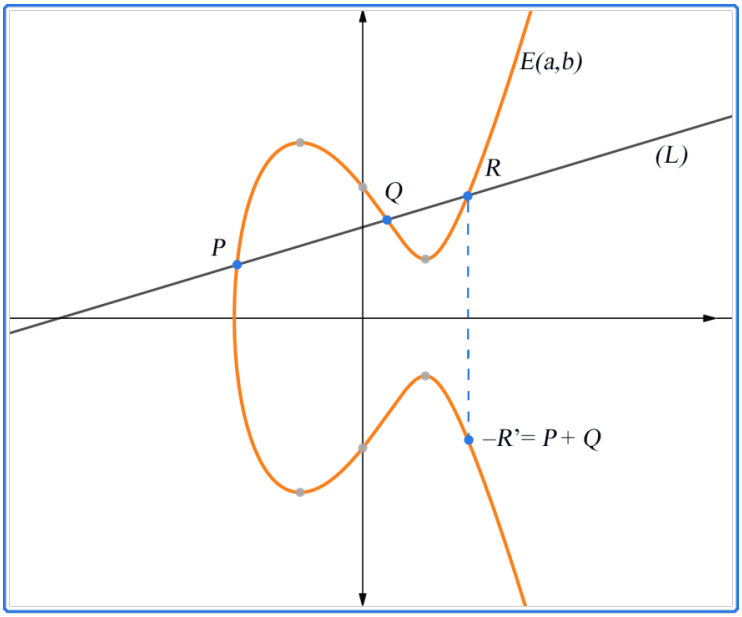

Figure 6.2 : La loi d'addition sur les courbes elliptiques.

- Maintenant, si nous voulons additionner un point P à lui-même, on prend simplement la droite (L) tangente à la courbe $E(a, b)$ en point P et qui coupe E en un autre point R. comme illustre la figure 6.3.

  Dans un certain sens, (L) coupe toujours $E(a, b)$ en trois points, donc P compte deux fois d'entre eux. On peut écrire que R' le symétrique du R par rapport à l'axe des abscisses (x, $–y$) est égale le doublement du point P ($R' = 2P$).

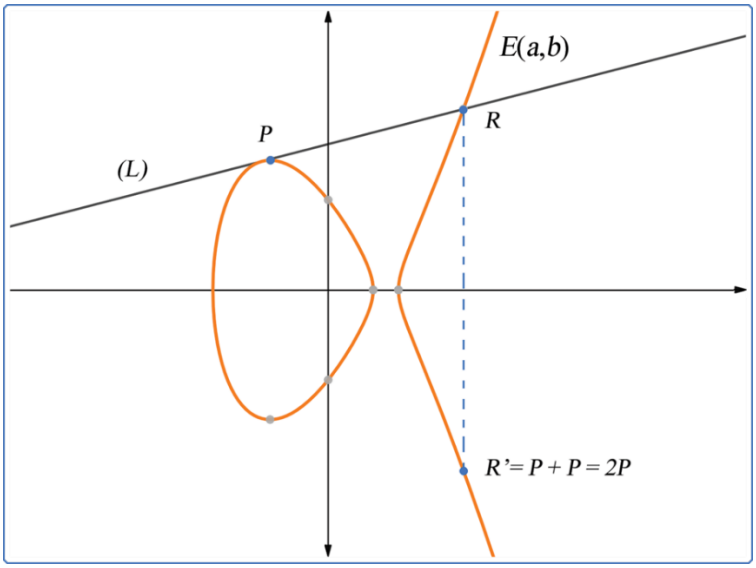

Figure 6.3 : L'addition de P à lui-même.

- Un deuxième problème potentiel se pose avec la "loi d'addition" réside au niveau de l'addition d'un point $P = (x, y)$ et de son point symétrique $P' = (x, -y)$ sur l'axe des abscisses [84]. La droite verticale (L) qui traverse P et P', qui coupe E en deux points seulement P et P' (Voir la figure 6.4).

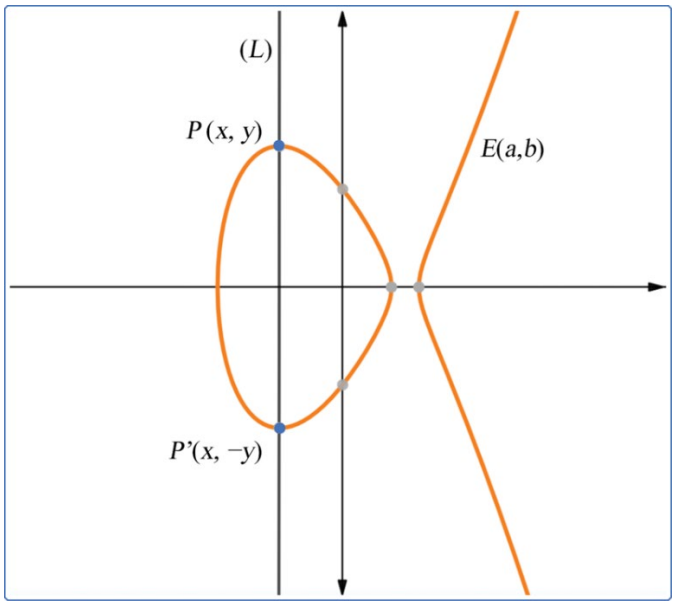

Figure 6.4 : La droite (L) parallèle à l'axe des ordonnées traversant deux point P et P'

Il n'y a pas de troisième point d'intersection, comme pour les cas précédents ce qui pose problème. La solution dans ce cas est de créer un extra-point O qui se situe "à l'infini". Plus précisément, le point O n'existe pas dans le plan x-y, mais nous prétendons qu'il se trouve sur chaque ligne verticale. Nous posons ainsi :

$$P \oplus P' = O$$

Nous devons également trouver comment additionner O avec un point ordinaire $P = (x, y)$ sur $E(a, b)$. Puisque O se trouve sur les lignes verticales, et cette ligne verticale coupe E aux points P, O et $P' = (x, -y)$ alors pour additionner P et O, nous allons faire la projection du point P' sur l'axe des abscisses, ce qui nous ramène à P. En d'autres termes, $P \oplus O = P$, donc O agit comme le zéro de l'addition pour les courbes elliptiques.

Nous définissons maintenant une courbe elliptique $E(a,b)$ par une équation de la forme.

$$\{(x, y)\in\mathbb{R}^2 \mid y^2 = x^3 + ax + b, 4a^3 + 27b^2 \neq 0\}\cup\{O\}.$$

6.3 Définition de groupe sur des courbes elliptiques

Un groupe sur des courbes elliptiques est définit comme suit :

- Les éléments du groupe sont les points d'une courbe elliptique;
- L'élément d'identité est le point à l'infini O;
- L'inverse d'un point P est celui symétrique autour de l'axe x;
- L'addition est donnée par la règle suivante:
  Étant donné trois points P, Q et R alignés et non nuls, leur somme est : $P + Q + R = O$.

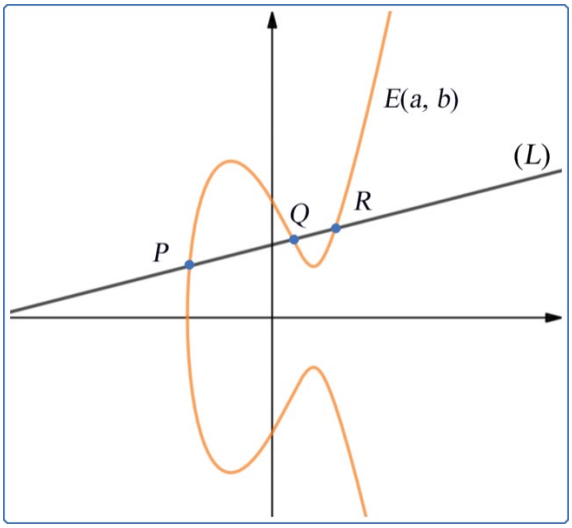

Figure 6.5 : Représentation graphique de la loi de groupe pour les courbes elliptiques

Nous avons besoin que de trois points alignés sans égard à l'ordre. Cela signifie que :

$$P + (Q + R) = Q + (P + R) = R + (P + Q) = \cdots = O$$

De cette façon, nous avons intuitivement prouvé que notre opérateur + est à la fois associatif et commutatif : nous sommes dans un groupe abélien.

La courbe elliptique arithmétique que nous avons décrite jusqu'à présent était sur des nombres réels. Ces courbes ne peuvent pas être utilisées telles quelles pour la cryptographie car les calculs avec des nombres réels sont arrondis à des erreurs. La cryptographie nécessite une arithmétique sans erreur. C'est après tout la raison principale de la notion de champ fini.

6.4 La courbe elliptique sur les champs finis

Les courbes elliptiques peuvent être définies sur des nombres réels, des nombres complexes, des nombres rationnels et des champs finis.

6.4.1 Définitions et propriétés

Un champ fini est d'abord un ensemble avec un nombre fini d'éléments. Un exemple de champ fini est l'ensemble des entiers modulo p, où p est un nombre premier. Il est généralement noté $\mathbb{Z}/p$, GF(p) ou $\mathbb{F}p$.

Une courbe elliptique sur champ fini GF(p) est l'ensemble des points décrits par l'équation :

$$E_p(a,b) = \{(x,y)\ \mathrm{GF}(p) \mid y^2 = x^3+ax+b \bmod p,\ 4a^3+27b^2 \neq 0\}\cup\{O\}.$$

- Ce qui était auparavant une courbe continue, est maintenant un ensemble de points disjoints dans le plan *x*-*y*. Mais nous pouvons prouver que même si nous avons restreint notre domaine, les courbes elliptiques en GF(*p*) forment toujours un groupe abélien.
- La figure 6.6 est l'ensemble des points de la courbe elliptique E_{109} (–9,0): $y^2 = x^3 – 9x \bmod 109$ sur un champ fini GF(109), avec $a = –9$, $b = 0$, $p = 109$.
- Notons qu'il y a au plus deux images pour chaque abscisse *x*. La distribution de ces points ayons une symétrie par rapport à la droite $y = p / 2 = 54,5$.

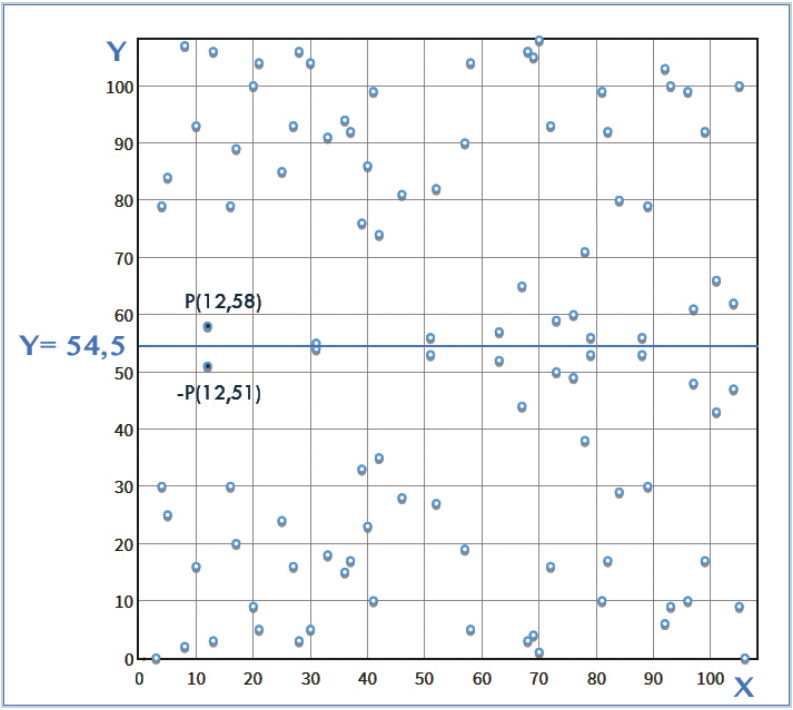

Figure 6.6 : L'ensemble des points de la courbe elliptique E_{109} (–9,0): $y^2 = x^3 – 9x \bmod 109$

Pour récapituler, un groupe abélien peut être défini comme suit :

- Les éléments du groupe sont les points d'une courbe elliptique E*p* (*a*, *b*);
- L'élément d'identité est le point à l'infini *O* (inf, inf);
- L'inverse d'un point *P* est celui symétrique autour de l'axe des *x*.

Supposons que $P(x, y)$, $Q(x_2, y_2) \in Ep(a, b)$, l'inverse de *P* est $–P$, $–P = (x, –y \bmod p)$ et $P + (–P) = O$, $Q – P = (x_2, y_2) – (x, y) = (x_2, y_2) + (x, –y \bmod p)$.

- L'addition est donnée par la règle suivante : étant donné trois points *P*, *Q* et *R* alignés et non nuls, leur somme est $P + Q + R = O$. Cette opération "d'addition" satisfait également la loi associative et commutative.

6.4.2 Addition et doublement des points de la courbe elliptique

- **Addition de deux points distincts :**

Pour tout $P(x_1, y_1)$ et $Q(x_2, y_2)$ de E*p* (*a*, *b*) avec $x_1 \neq x_2$, $P + Q = (x_3, y_3)$ est défini comme suite :

$$x_3 = \lambda^2 – x_1 – x_2 \bmod p \text{ avec } \lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} \bmod p$$
$$y_3 = \lambda (x_1 – x_3) – y_1 \bmod p$$

- **Doublement d'un point :**

Pour tout $P(x_1, y_1) \in Ep(a,b)$ avec $y_1 \neq 0$, $2P = (x_2, y_2)$ est défini comme:

$$x_2 = \lambda^2 – 2x_1 \qquad \text{avec} \quad \lambda = \frac{(3x_1^2 + a)}{2y_1}$$

$$y_2 = \lambda\,(x_1 - x_2) - y_1$$

6.4.3 Optimisation du doublement d'un point sur une courbe elliptique

Comme pour les réels, la multiplication peut être définie comme :

$$nP = \underbrace{P+P+\cdots+P}_{n \text{ opérations}}$$

De nombreux auteurs ont discuté des méthodes d'exponentiation dans un groupe multiplicatif, qui peuvent donc être étendues au calcul de la multiplication scalaire elliptique [87].

La méthode la plus simple pour calculer nP est basée sur la représentation binaire de n. Si n a k chiffres binaires, c'est-à-dire $n = b_{k-1} \ldots b_1 b_0$, la complexité temporelle de l'algorithme serait $O(2^k)$, ce qui n'est pas vraiment bon. Par conséquent, nous proposons un algorithme plus rapide en utilisant des opérations de doublement et d'addition. Si n est représenté sous forme binaire, cette représentation peut être transformée en une somme de puissances de deux (comme dans l'équation (2).

$$n = \sum_{i=0}^{k-1} b_i\, 2^i \qquad \text{avec } b_i \in \{0,1\} \qquad (2)$$

Il semble que le calcul de nP nécessite $(n–1)$ additions, sa complexité temporelle est $O(n)$.
Son principe de fonctionnement peut être mieux expliqué par un exemple. Prenons $n = 27$. Sa représentation binaire est $(11011)_2$. Cette représentation peut être transformée en une somme de puissances de deux : $27 = 2^0 + 2^1 + 2^3 + 2^4$ (nous avons pris chaque chiffre binaire de n et nous l'avons multiplié par une puissance de deux d'après (2)) $27P = 2^0P + 2^1P + 2^3P + 2^4P$.

L'algorithme d'écrit les étapes d'addition et de doublement pour calculer $27P$

- Prendre P;
- Doublez-le, nous obtenons $2P$;
- Ajoutez $2P$ à P (nous obtenons le résultat de $2^0P + 2^1P$);
- Double $2P$, nous obtenons 2^2P;
- Double 2^2P, nous obtenons 2^3P;
- Ajoutez-le à $(2^0P + 2^1P)$, nous obtenons $2^0P + 2^1P + 2^3P$;
- Double 2.2^3P, nous obtenons 2^4P;
- Ajoutez-le à notre résultat (pour obtenir $2^0P + 2^1P + 2^3P + 2^4P$).

Au final, nous calculons $27P$ en effectuant seulement quatre doublements et trois additions. Voici un script Python qui implémente notre algorithme proposé :

Algorithme optimisé pour calculer un multiple d'un point sur une courbe elliptique (script python)
1 Def bits(n): Generates the binary digits of n, starting from the least significant bit. 2 Returns the result of $n \times P$ computed using the double and add algorithm:

```
while n:
    yield n &1
    n>>=1
def double_and_add(n, P)
result=0
addend= P
for bit in bits(n):
    if bit ==1:
        result+= addend ''call operation of addition of
        points''
addend*=2  ''call multiples of point ''
        return result
```

6.5 Le sous-groupe, l'ordre et le générateur d'un groupe de courbe elliptique

6.5.1 Le sous-groupe

Un "sous-groupe" est un groupe qui est un sous-ensemble d'un autre groupe.
Un "sous-groupe cyclique" est un sous-groupe dont les éléments sont répétés cycliquement : On prend un point générique G, on exécute circulairement n fois multiple du point G jusqu'à $nG = O$. On obtient G, $2G$, .., $(n-1)G$ et O, au total n points de courbe elliptique, qui forme un cyclique sous-groupe basé sur le point G. Le point G est appelé générateur ou point de base de ce sous-groupe cyclique, dont son ordre est le nombre naturel n.

6.5.2 l'ordre d'un groupe de courbe elliptique

Nombre de points N dans une courbe elliptique définie sur un champ fini : nous avons dit qu'une courbe elliptique définie sur un champ fini a un nombre fini de points. Une question importante à laquelle nous devons répondre est : combien de points y a-t-il exactement ?.
Tout d'abord, disons que le nombre de points dans un groupe est appelé l'ordre du groupe noté N. Essayons toutes les valeurs possibles pour x de 0 à $p-1$ n'est pas un moyen possible de compter les points, car cela nécessiterait $O(p)$ opération, et c'est "dur" si p est un grand premier. Il existe un algorithme plus rapide pour calculer l'ordre en utilisant l'algorithme Schoof [85]. Il s'exécute en temps polynomial par contre le théorème de Hasse [85] fournit des limites plus strictes pour N. Prenons cet exemple : Soit $E_{109}(-9,0) = x^3 - 9x \bmod 109$. Calculons l'intervalle de Hasse de E_{109} : $p+1-2\sqrt{p} \le N \le p+1+2\sqrt{p}$ avec $p = 109$,

$$90 \le N \le 130$$

On peut montrer que le point (8, 2) a un ordre $n = 52$, donc N est un multiple de n, et les multiples de n sont 52, 104, 156,… d'où $N = 104$ d'après l'inégalité de Hasse. Donc la courbe elliptique $E_{109}(-9,0)$ contient 104 points y compris le point à l'infini O.

Les sous-groupes cycliques d'une courbe elliptique sont les fondements d'une cryptographie à courbe elliptique et d'autres crypto-systèmes. De plus, il faut sérieusement choisir le sous-groupe cyclique d'ordre n. Pour que l'algorithme de cryptographie à courbe elliptique fonctionne correctement, n doit être un nombre premier et diviseur de N.

6.5.3 Sélection d'un point générateur approprié pour générer un sous-groupe cyclique

Pour la mise en œuvre d'un crypto-système à courbe elliptique, il est nécessaire de générer un sous-groupe cyclique du groupe de points sur la courbe elliptique.

Pour choisir un point générateur approprié pour générer le sous-groupe cyclique, il convient d'utiliser l'algorithme suivant :

1) Calculer l'ordre N de la courbe elliptique.
2) Parmi les diviseurs de N, choisissez le plus grand diviseur premier comme n. Alors n sera l'ordre du sous-groupe cyclique à générer.
3) Calculer $h = \frac{N}{n}$. (Selon le théorème de Lagrange [97], h est toujours un entier. h est connu comme le cofacteur du sous-groupe).
4) Sélectionner un point aléatoire P sur la courbe elliptique.
5) Calculer $G = hP$.
6) Si $G = O$, il faut retourner à l'étape 4). Sinon, G est le point générateur approprié du sous-groupe cyclique.

L'algorithme ci-dessus ne fonctionne que si n est un nombre premier. Si n n'est pas un nombre premier, l'ordre de G ne pourrait être qu'un de ses multiples.

Par exemple, la courbe $y^2 = x^3 - 2x + 2$ sur le champ E_{31} (–2,2) a l'ordre $N = 30$. Ses sous-groupes peuvent avoir l'ordre de tous les diviseurs de N, c'est-à-dire $n = 1, 2, 3, 5, 6, 10, 15$ ou 30. Nous calculons $h = N/n = 30/5 = 6$, si nous choisissons au hasard quelques points de la courbe elliptique $\{P = (1,30), (9,0), (6,12), (18,1), (23,23), ..\}$, nous pouvons voir que $hP = 6(1,30) = O$, $6(9,0) = O$, $6(6,12) = O$, $6(18,1) = (18,1) \neq O$, $6(23,23) = (23,23) \neq O$.

On peut dire que les points $G(18,1)$ et $G(23,23)$ sont des générateurs du sous-groupe cyclique d'ordre 5. Pour trouver les autres points, nous choisissons $G(18,1)$ et calculons: $1G(18, 1) = (18,1)$, $2G(18,1) = (23,23)$, $3G(18,1) = (23,8)$, $4G(18,1) = (18, 30)$, $5G(18,1) = O$, $6G = (18,1)$, $7G = (23,23)$ .., le résultat est toujours l'un de ces cinq points, qui forme un sous-groupe cyclique basé sur G (voir la figure 39). L'ordre des sous-groupes est 5.

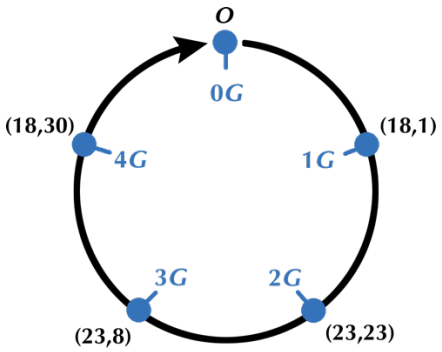

Figure 6.7 : Le point générateur $G(18,1)$, qui se répète cycliquement

6.6 Cryptographie à courbe elliptique

Les paramètres suivants sont les paramètres de domaine spécifiques requis pour faire fonctionner n'importe quel algorithme sous ECC.

- Le premier p qui spécifie la taille du champ fini.
- Les coefficients a et b de l'équation de la courbe elliptique $E(a, b)$.
- Le point de base G qui génère le sous-groupe cyclique.
- L'ordre n du sous-groupe.
- Le cofacteur h du sous-groupe.

Dans le cryptosystème à courbe elliptique :

1) La clé privée est un entier aléatoire k choisi parmi {1, 2,.., (n-1)}.

2) La clé publique est le point $P_{pu} = kG$.

Si k et G sont connus (avec les autres paramètres de domaine), le calcul de P_{pu} est "facile". Mais si seuls *Ppu* et G sont connus, trouver la clé privée k est "difficile" car cela nécessite de résoudre le problème ECDLP (Elliptic Curve Discrete Logarithm).

6.6.1 Problème de Logarithme discret de courbe elliptique (PLD)

Ce problème s'exprime comme suit : étant donné une courbe elliptique E sur un corps Fp, soit P un point générateur du groupe sur E et un autre point Q de la courbe. Comment trouver le plus petit entier n tel que Q=n.P ?

Ce problème, connu sous le nom du **problème de logarithme discret (PLD)** pour les courbes elliptiques, est considéré comme un problème "difficile", car il n'existe aucun algorithme de temps polynomial connu pouvant fonctionner sur un ordinateur classique. Il n'y a cependant aucune preuve mathématique de cette croyance.
Le PLD est le même qu'avec d'autres cryptosystèmes tels que l'algorithme de signature numérique (DSA), l'échange de clés Diffie-Hellman (DH) et l'algorithme ElGamal, ce n'est pas un hasard s'ils ont le même nom. La différence est qu'avec ces algorithmes, nous utilisons l'exponentiation modulo au lieu de la multiplication scalaire. Leur problème de logarithme discret peut être énoncé comme suit: si nous savons a et b, c'est quoi k? Tel que $b = a^k \bmod n$.

Ces deux problèmes sont "discrets" car ils impliquent des ensembles finis (plus précisément, des sous-groupes cycliques). Et ce sont des "logarithmes" car ils sont analogues aux logarithmes ordinaires. Le tableau 11 répertorie trois types des cryptosystèmes et leurs problèmes de sécurité :

Système cryptographique	Problème mathématique	La description	Temps de vitesse
ECC, ECDH	Logarithme discret à courbe elliptique	Étant donné une courbe elliptique E et les points P et Q sur E, trouver n tel que $Q = n.P$	Exponentiel
RSA	Factorisation entière	Étant donné un nombre n, trouver ses facteurs premiers p et q	Sous-exponentielle
Elgamal, DSA, DH	Logarithme discret	Étant donné un nombre premier n, et les nombres g et h, trouver x tel que $h = g^x \bmod n$	Sous-exponentielle

Tableau 6.1 : Cryptosystèmes à clé publique et leurs problèmes mathématiques.

Ce qui rend ECC intéressant, c'est qu'à partir d'aujourd'hui, le problème du logarithme discret pour les courbes elliptiques semble être "plus difficile" par rapport à d'autres problèmes similaires utilisés en cryptographie. Cela implique que nous avons besoin de moins de bits pour l'entier afin d'atteindre le même niveau de sécurité qu'avec les autres cryptosystèmes, comme nous le verrons en détail par la suite dans cette section.

6.6.2 Analyse de performance de RSA/DSA et ECC

Pour mieux évaluer l'efficacité de ECC, il est nécessaire d'établir des comparaisons avec les systèmes RSA et DSA (par exemple). Ces comparaisons vont porter essentiellement sur les niveaux de sécurité, et lorsqu'on parle de l'efficacité d'un système cryptographique à clé publique, il y a trois facteurs à considérer :

- Les couts de calcul nécessaire pour effectuer la génération des clés publiques et privées.
- La Taille des clés en bits nécessaires pour stocker les paires de clés et les paramètres système associés.
- Le nombre de bits à communiquer pour transférer un message crypté ou une signature (bande passante).

Commençons par le tableau 6.2 qui présente la comparaison des longueurs de clé des systèmes RSA/DSA et ECC, la première colonne du tableau indique le temps d'estimation pour rompre les deux systèmes de cryptage/décryptage en années MIPS (ou en Processeurs MIPS).

Année MIPS (Million d'Instructions Par Seconde)	**RSA/DSA Taille des clés (bit)**	**ECC Taille des clés (bit)**
10^4	512	-
10^8	768	132
10^{11}	1024	160
10^{20}	2048	210
10^{78}	21000	600

Tableau 6.2 : Estimation du niveau de sécurité des ECC et RSA/DSA en années MIPS

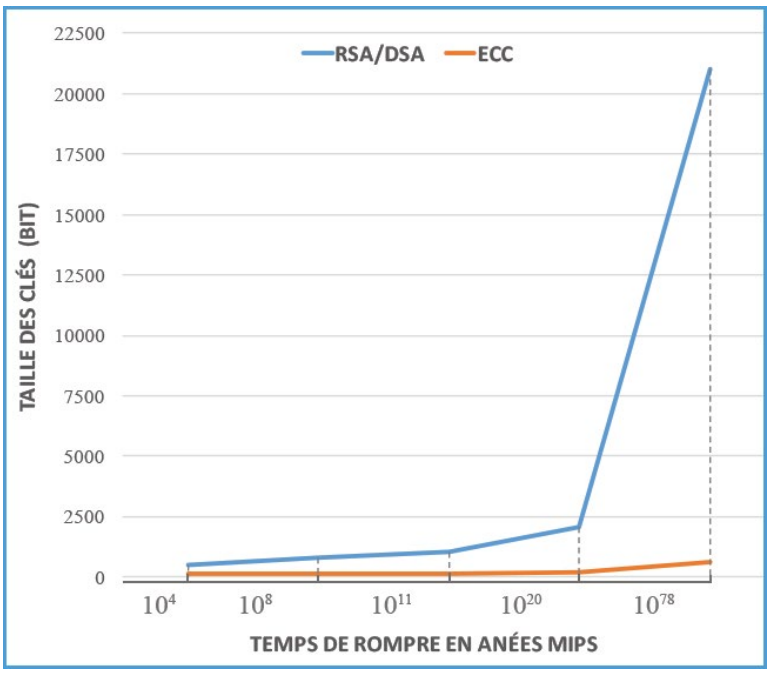

Figure 6.8 : Estimer le niveau de sécurité des ECC et RSA en années MIPS

Remarquons que le rapport des clés sont très diffèrent entre les deux systèmes. Comme la puissance de calcul des machines augmente très rapidement, les tailles des clés doivent aussi augmenter si on veut garder le même niveau de sécurité. Donc le tableau 6.2 prouve que ceci est irréalisable avec RSA pour un matériel disposant de ressources limitées, c'est pourquoi un ECC semble être idéal dans cette évolution.

Remarque : A. Shamir [88] décrit une nouvelle implémentation matérielle qui améliore le temps d'exécution de trois à quatre ordres de grandeur par rapport aux implémentations actuelles (nouvelles ressources en hardware). Avec ce matériel, Shamir estime que la prise en compte du RSA 512 bits peut être brisé en 10 minutes par un appareil à 10 000 $ et RSA 1024 bits en moins d'un an avec un appareil à 10 millions de dollars.

6.6.3 Comparaison entre les temps de calcul

Pour mieux évaluer l'efficacité des ECC, on va comparer les temps de calcul en millisecondes pour la génération des clés, signature et sa vérification pour le schéma ECDSA (dérivé du DSA, utilisant les courbes elliptiques) aux temps correspondants pour RSA [89].

	ECDSA (ms)	RSA (ms)	DSA (ms)
Génération des clés	5,5	10^3	22,7
Signature	6,3	43,3	23,6
Vérification	26	0,65	28,3
Total (ms)	37,8	1043,95	74,6

Tableau 6.3 : Comparaison des temps de calcul

D'après le résultat global montré dans la figure 6.9, Il est clair que la signature basée sur ECC serait beaucoup plus efficace en termes de sécurité avec des tailles de clé beaucoup plus courtes, et comme ECC offre le même niveau de sécurité avec une taille de clé beaucoup plus petite, il peut être un système fiable pour la cryptographie asymétrique.

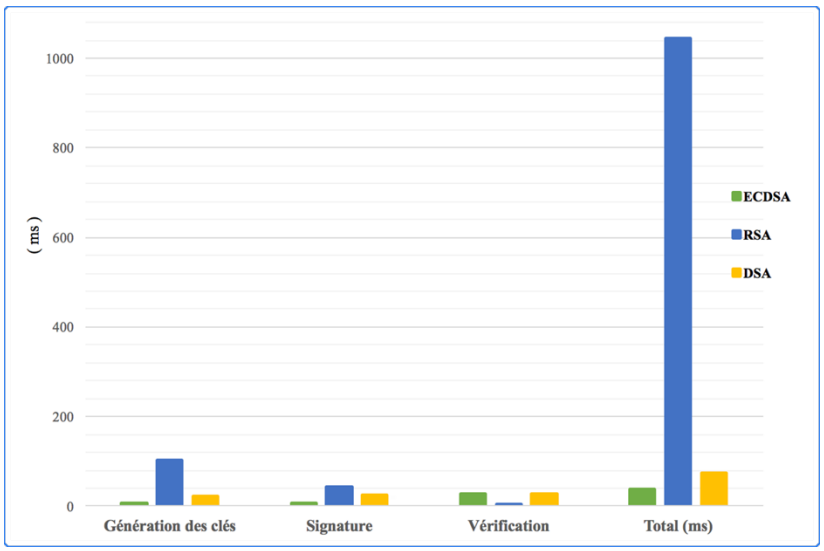

Figure 6.9 : Comparaison des performances de ECDSA-160, RS-1024 et DS-1024.

6.6.4 Cartographie (Mapping) des codes de texte en clair aux points d'une courbe elliptique

6.6.4.1 La méthode de cartographie (Mapping) de Koblitz

ECC est l'un des outils de cryptographie les plus aboutis et les plus largement utilisés, mais les moins bien compris [80], car il s'agit d'une technique de cryptographie à clé publique qui repose sur l'arrangement algébrique de courbes elliptiques sur des champs finis.

Plusieurs articles proposent des méthodes de mappage du message d'origine en une valeur numérique en effectuant certaines opérations mathématiques. Le codage du message en texte

brut sur la courbe elliptique n'est pas très simple car il n'y a pas d'algorithme déterministe dans le temps polynomial pour trouver les points sur la courbe elliptique.
Nous décrivons la méthode de cartographie de Koblitz pour un message en texte brut à un point EC où la courbe elliptique Ep est définie sur un champ fini.

$$\mathrm{E}p\ (a, b): y^2 = x^3 + ax + b \mod p$$

Supposons que M est un message en texte brut. De toute évidence, quel que soit le message, M peut être interprété comme une chaîne binaire $M = m_1 m_2 ... m_n$, où $m_i = b_{\ell-1} ... b_1 b_0$ représente un bloc de texte binaire de longueur ℓ.

Essayons de mapper chaque bloc m_i dans les points de la courbe elliptique E:

$$m_i \longrightarrow Pm_i = (x_j, y_i) = ((m_i \times r + j) \bmod p, \sqrt{x_j^3 + ax_j + b})$$

Soit r un entier, et $0 \leq j < (r - 1)$ et $x_j = (m_i \times r + j) \bmod p$.

On calcule $y_j^2 = x_j^3 + ax_j + b$, et on vérifie si y_j a une racine carrée mod p, si c'est le cas on s'arrête. Sinon, on continue de calculer le prochain x_j. Ce processus continue, il ne s'arrête que lorsqu'une racine carrée $Pm_i(x_j, y_i)$ est trouvée, dont laquelle nous avons réussi le mappage. Cette méthode est généralement attribuée à Koblitz.
Pour récupérer le message m_i, on calcule $m_i = \lfloor \frac{x_j}{r} \rfloor$, où $\lfloor \frac{x_j}{r} \rfloor$ désigne le plus grand entier inférieur ou égal à $\frac{x_j}{r}$.

Prenons un cryptage basé sur la courbe elliptique suivante : E_{109} (–9,0): $y^2 = x^3 - 9x \bmod 109$, et choisissons un bloc de texte $m_i = 18$, avec $r = 5$ et $j = 2$, ce qui donne $x_j = 18 \times 5 + 2 = 92 \bmod 109 = 92$, et $y_j^2 = (923 - 9 \times 92) \bmod 109 = 777860 \bmod 109$, $y_j = \sqrt{36} = 6$. Donc le bloc en texte clair m_i est mappé au point (92, 6) de la courbe elliptique E_{109} (–9,0), et $m_i = \lfloor \frac{92}{5} \rfloor = 18$.
La probabilité que l'algorithme de mappage réussit à mapper M à un point EC est $(1 - \frac{1}{2^r})$, c'est-à-dire que le codage peut être effectué (plus probablement) lorsque l'entier r est grand.
Pour que le codage/décodage fonctionne correctement, nous devons choisir un r qui vérifie l'inégalité de Koblitz $(m_i + 1) \times r < p$.
L'algorithme de Koblitz est officiellement présenté ci-dessous (Algorithm1).

Algorithme 1 de Koblitz : Mappage probabiliste d'un message en clair sur un point d'une courbe elliptique définie sur un champ fini

for j = 0 to r–1 **do**
  let $x_j = m \times r + j$ mod p
  if $y_j = x_j^3 + ax_j + b$ has a square root mod p
    break
  end if
end for
if $j < r$ **then**
  compute a square root of y_j mod p
  map m to (x_j, y_j)
else
  output "unsuccessful in the attempt to map M to an EC point"
end if

6.6.5 L'algorithme de cartographie de Koblitz modifié

Notre modification est porté sur les boucles des variables m_i des blocs du message $M = m_1m_2 ... m_n$ dans l'algorithme de Koblitz. Nous avons fixé trois valeurs: le bloc maximal m_{max} et le minimal m_{min} de M et la constante r_{max}.
Pour que l'algorithme de Koblitz soit plus avantageux, nous proposons trois modifications:

1) Selon l'inégalité de Koblitz $(m_i + 1) \times r < p$, nous pouvons déterminer la valeur maximale r_{max} de r qui maintient l'inégalité juste, nous choisissons donc la boucle de r_{max} jusqu'à 1.

2) Pour chaque valeur de r, recherchons l'existence des racines carrées de l'équation EC correspondant aux blocs m_i du message M, avec m_{min} correspondant à la valeur minimale en ASCII du bloc de message M et m_{max} sa valeur maximale.

3) Si le message M contient des blocs en double, nous codons un seul bloc parmi les autres.

L'algorithme devient :

Algorithme 2 de Koblitz modifié : Cartographie modifiée probabiliste d'un message en texte clair à un point sur une courbe elliptique définie sur un champ fini

for $r = r_{max}$ to 1 **do**
for m_{min} to m_{max} **do**
for j=0 to (r_{max}–1)
let $x_j = m_i \times r_k + j$ mod p **end if**
if $y_j = x_j^3 + ax_j + b$ has a square root mod p
Test the existence of (x_j, y_j) in the list of points
if yes do not do anything
if not write (x_j, y_j)
break
end if
end for
if j<r **then**
compute a square root of y_j mod p
map m_i to (x_j, y_j)
else
output "unsuccessful in attempt to map M to an EC point"

Nous avons traduit l'algorithme modifié de Koblitz en une interface utilisateur graphique développée en langage Java, pour mapper un message M en ℓ bits. L'interface graphique a été conçue pour permettre à l'utilisateur de saisir les paramètres a, b, p d'une courbe elliptique, et les paramètres r_{max},

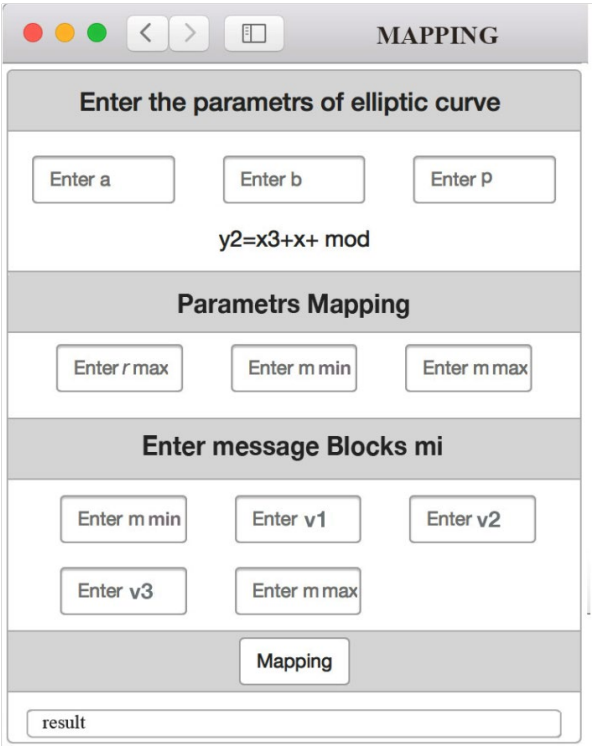

Figure 6.10 : Interface émetteur (Cartographie/Mapping)

Notez que les valeurs v_1, v_2 et v_3 (voir figure 6.10) ne sont que des valeurs ASCII des blocs de message $M = m_{min}..v_1v_2v_3..m_{max}$ qui sont bornées entre les blocs m_{min} et m_{max}. L'ajout d'autres valeurs v_4, v_5 .., dépend bien évidement de la longueur du message M. Si l'algorithme 2 ne donne pas les points correspondant aux blocs de message M, on change l'équation de la courbe elliptique.

6.7 Le chiffrement, déchiffrement et décodage utilisant la méthode de Koblitz modifié

6.7.1 Algorithme de chiffrement utilisant ECC

Aicha veut envoyer un message M à Brahim. Ils s'accordent sur une courbe elliptique : Ep (a, b): $y^2 = x^3 + ax + b \bmod p$ (ils font des communications confidentielles entre eux). Avant de passer au chiffrement du message M, Aicha mappe chaque bloc de texte en clair m_i du message M, en Pmi (x_j, y_j) par la méthode de mappage indiquée dans la section précédente (6.3.2).

Le processus de chiffrement de Aicha est décrit ci-dessous :

- **Étape 1:** Aicha choisit une clé secrète k_A parmi $\{1,.., (n–1)\}$; elle calcule $P_A = k_AG$.
- **Étape 2 :** Elle sélectionne la clé publique de Brahim $P_B = k_BG$ et calcule la clé commune k_AP_B.
- **Étape 3:** Elle calcule $P_C = Pm_i(x_j, y_j) + k_AP_B$.
- **Étape 4:** Aicha envoie maintenant le message chiffré $\{r, P_A, P_C\}$ à Brahim.

6.7.2 Algorithme de déchiffrement utilisant ECC

Le processus de décryptage de Brahim est décrit ci-dessous :

- **Étape 1:** Brahim sélectionne la clé publique de Aicha P_A et calcule le k_BP_A.
- **Étape 2:** Il calcule $P_C–k_BP_A = Pm_i\ (x_j, y_j) + P_B–k_BP_A = Pm_i\ (x_j, y_j) + k_BP_A–k_BP_A = Pm_i\ (x_j,$

y_j).

6.7.3 Décodage

Après déchiffrement, le destinataire obtient le message m_i à partir du point Pm_i (x_j, y_j) en utilisant la formule suivante : $m_i = \lfloor x_j / r \rfloor$, qui est le plus grand entier inférieur ou égal à x_j / r. De même, nous avons également développé une application de décodage avec l'interface suivante :

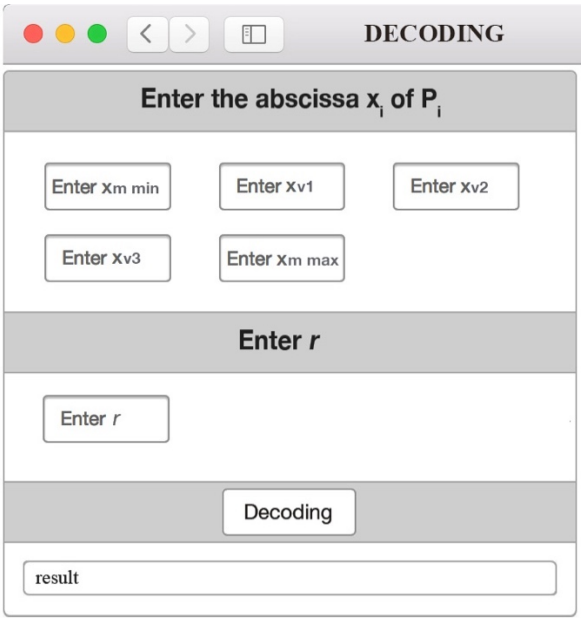

Figure 6.11: Interface du récepteur (décodage)

6.8 La langue amazighe et son système d'écriture.

Avant de présenter une implémentation sur le cryptage ECC, nous aimerons utiliser un message écrit en alphabet Tifinagh en utilisant le codage Unicode. L'alphabet Tifinagh, adopté par l'Institut royal de la culture amazighe (IRCAM), a été officiellement reconnu par l'Organisation internationale de normalisation (ISO) sur la norme ISO / CEI 10646 en 2003 comme le plan multilingue de base [92]. La pla ge de points de code réservée aux amazighs occupe l'espace hexadécimal de 2D30 à 2D7F; il contient ainsi 30 alphabets Tifinagh.

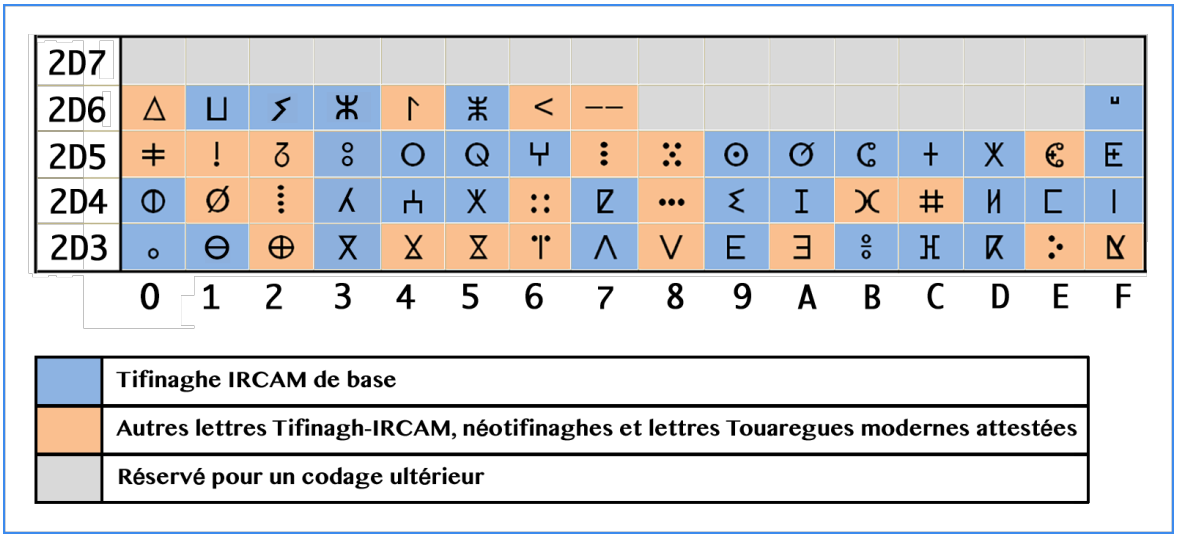

Figure 6.12. : Système d'écriture en langue amazighe Unicode en Afrique du Nord

6.9 Implémentation de chiffrement à l'aide des courbes elliptiques

Supposons que la cryptographie à courbe elliptique soit basée sur E_{4019} (–10,16): $y^2 = x^3 – 10x + 16 \bmod 4019$ sur un champ fini GF(4019). Nous calculons l'ordre N de la courbe elliptique en utilisant l'algorithme de Schoof [85], nous obtenons $N = 3912$, y compris le point à l'infini.
Selon la section "**6.5.2**" on choisit l'ordre du sous-groupe $n = 163$, $h = N / n = 3912/163 = 24$, un point générateur $G = (885,527)$ du sous-groupe, de l'ordre 163 (nombre premier) est sélectionné.
Les paires de clés privées de chiffrement et de déchiffrement peuvent être sélectionnées parmi {1, 2, ..., 162}.

Prenons un message écrit en caractère tifinaghe en clair M est " ⵙⵅⵚⵉ " (*Bonjour*), il est codé en binaire comme dans ce tableau :

Caractères du message M	Representation Hexadecimal	Conversion binaire
ⵙ	2D30	10110100110000
ⵅ	2D63	10110101100011
ⵚ	2D53	10110101010011
ⵉ	2D4D	10110101001101

Tableau 6.4 : Conversion du message M en binaires.

M =1011010 0110000 1011010 1100011 1011010 1010011 1011010 1001101
Aicha veut envoyer le message M = "ⵙⵅⵚⵉ" à l'utilisateur Brahim, elle divise le message en 7 bits.

Positions des blocs m_i	Representation desblocs m_i en binare	Representation desblocs m_i en décimal
1	1011010	90
2	0110000	48
3	1011010	90
4	1100011	99
5	1011010	90
6	1010011	83
7	1011010	90
8	1001101	77

Tableau 6.5 : Conversion des blocs de messages en décimal

Les blocs de messages $M = m_1m_2m_3m_4m_5m_6m_7m_8$ contiennent des doublons : $m_1=m_3=m_5=m_7=90$, $m_2 = 48$, $m_4 = 99$, $m_6 = 83$, $m_8 = 77$, la cartographie /mapping sera sur cinq blocs au lieu de huit bloc.

6.9.1 Mapping

Aicha utilise les paramètres de l'algorithme 2, elle sélectionne le plus petit bloc $m_{min} = 48$, le plus grand bloc $m_{max} = 99$ et $r_{max} = 40$ selon l'inégalité de Kolbitz: $(m_{max} + 1) \times r_{max} < 4019$.

Les blocs à mapper sont : $m_{min} = 48$, $m_8 = 77 = v1$, $m_2 = 83 = v_2$, $m_3 = 90 = v_3$, $m_{max} = 99$.

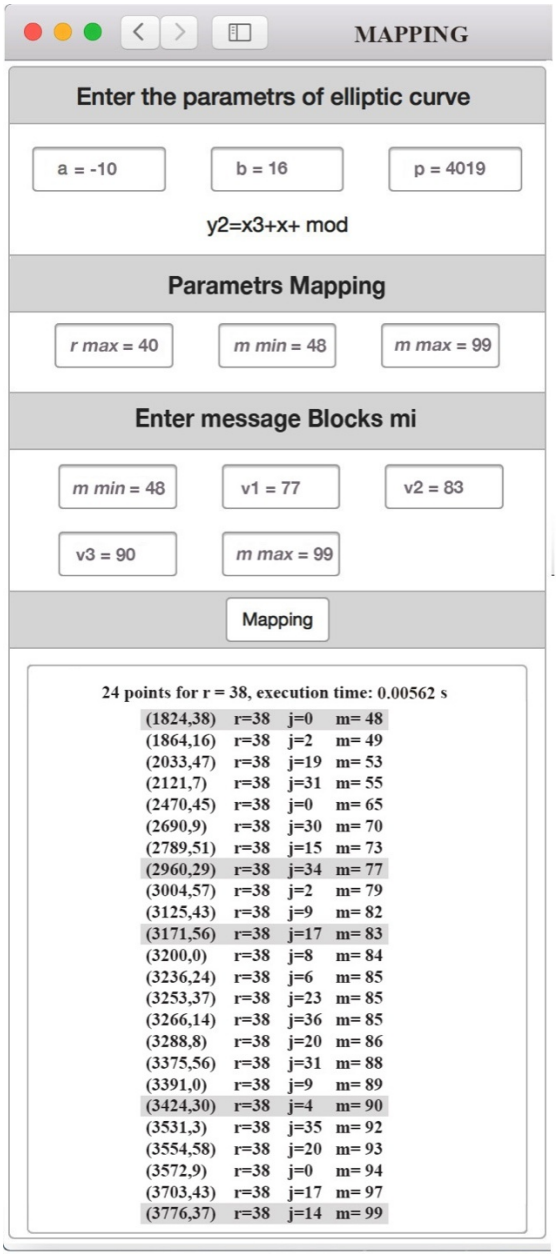

Figure 6.13 : Insertion des paramètres de codage et de leurs résultats du message *M*.

L'algorithme proposé à trouver le paramètre $r = 38$ qui correspond aux blocs ASCII du message *M*, pour un temps d'exécution de 0,00562 seconde. Le mapping de Aicha est représenté dans le tableau 6.6:

Positions des Blocs m_i	**Representation des Blocs m_i en binaire**	**Representation des Blocs m_i en décimal**	**Mapping des$m_i$aux point Pm_i (x_i, y_i), Avec r=38**
1	1011010	90	Pm_1(3424, 30)
2	0110000	48	Pm_2(1824, 38)
3	1011010	90	Pm_3(3424, 30)
4	1100011	99	Pm_4(3776, 37)
5	1011010	90	Pm_5(3424, 30)
6	1010011	83	Pm_6(3171, 56)

7	1011010	90	Pm_7(3424,30)
8	1001101	77	Pm_8(2960, 29)

Tableau 6.6 : Mapping des blocs m_i de messages M en points Pm_i (x_j, y_j).

6.9.2 Chiffrement

Aicha et Brahim ont choisis secrètement leurs clés privées parmi {1,..,162}, k_A= 7 et k_B = 5

• **Étape 1:** Aicha calcule sa clé publique $P_A = k_A G$=7(885, 527) = 7P_A= (610, 1126)

• **Étape 2:** Elle sélectionne la clé publique de Brahim $P_B = k_B G$=5(885, 527) = (2149, 1142) pour calculer la clé publique commune $k_A P_B = k_B P_A$ = (242, 3367).

• **Étape 3:** Elle additionne $k_A P_B$ pour chaque point Pm_i, donc $P_{Ci} = Pm_i(x_j, y_j) + k_A P_B$ avec i = {1,..,8}(tous les résultats de chiffrement/déchiffrement sont affichés dans le tableau 6.12).

• **Étape 4:** Aicha envoie maintenant le message chiffré {r, P_A, P_{Ci}} à Brahim.

6.9.3. Déchiffrement

Le processus de décryptage par Brahim est décrit ci-dessous pour un point.

• **Étape 1**: Brahim sélectionne la clé publique de Aicha P_A et calcule $k_B P_A$ = 5 (610, 1126) = (242, 3367).

• **Étape 2:** Il calcule: $P_{C1} - k_B P_A$ = (3858, 2910) – (242, 3367) = (3424, 30) = Pm_1.

6.9.4 Décodage.

Après le décryptage des points envoyés par Aicha, Brahim entre les abscisses x_j des cinq points (pas de doublons) Pm_i dans l'application de décodage pour trouver les blocs m_i.

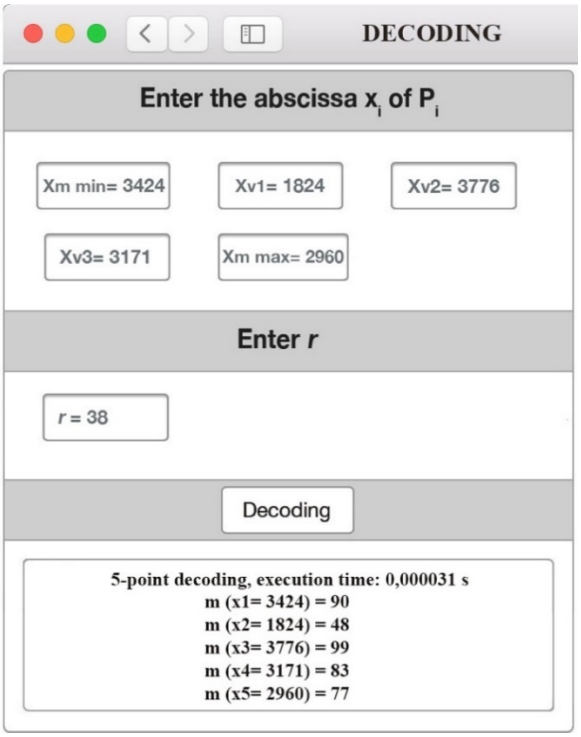

Figure 6.14 : Insertion des paramètres de décodage du message M et de leurs résultats.

Caractères de message *M*	**Mapping m_i en point $Pm_i(x_j, y_j)$, avec $r = 38$**	**Chiffrement:** $k_A = 7$ $P_A = (610, 1126)$ $k_AP_B = (242, 3367)$ $P_{Ci} = Pm_i(x_j, y_j) + k_AP_B$, $\{ r = 38, P_A, P_{Ci} \}$ $Pm_i = P_{Ci} - k_AP_B$	**Dechiffrement :** $k_B = 5$ $P_B = (610, 1126)$ $k_BP_A = (242,3367)$ $Pm_i = P_{Ci} - k_AP_B$	**Decoding:** $m_i = \lfloor x_j / r \rfloor$	**Caractères de message *M***
ⴰ	Pm_1(3424, 30)	{38, (610, 1126), (1765, 851)}	Pm_1 (3424, 30)	90	ⴰ
	Pm_2(1824, 38)	{38,(610, 1126), (3226, 3212)}	Pm_2 (1824, 38)	48	
ⵊ	Pm_3(3424, 30)	{38,(610, 1126), (3858, 2910)}	Pm_3 (3424, 30)	90	ⵊ
	Pm_4(3776, 37)	{38,(610, 1126), (1758, 3336)}	Pm_4 (3776, 37)	99	
ⵙ	Pm_5(3424, 30)	{38, (610, 1126), (1765, 851)}	Pm_5 (3424, 30)	90	ⵙ
	Pm_6(3171, 56)	{38, (610, 1126), (910, 1285)}	Pm_6 (3171, 56)	83	
ⵉ	Pm_7 (3424,30)	{38, (610, 1126), (1765, 851)}	Pm_7 (3424,30)	90	ⵉ
	Pm_8(2960, 29)	{38,(610, 1126), (1327, 3197)}	Pm_8 (2960, 29)	77	

Figure 6.7 : cartographie, chiffrement, déchiffrement et décodage pour le message M = ⴰⵊⵙⵉ.

6.10 Travaux connexes

Quelques méthodes de codage/décodage d'un message qui nous intéresse, utilisant des courbes elliptiques sur un champ fini. La méthode proposée par Amounas dans [90] est basée sur les points générateurs et sur une matrice pour changer la position des points de CE :

Soit Ep (a, b) une courbe elliptique choisie par Aicha avec un point générateur G. Elle mappe les alphabets a à G, b à $2G$, c à $3G$ (points publiques). Le message se termine par un espace ((0,0) sur Ep (a, b) de sorte que la taille du message soit un multiple de 3, et la disposition des points du message forme une matrice tridimensionnelle appelée M. Puis, elle sélectionne une matrice A non singulière de telle sorte que det (A) = ±1 pour calculer $Q = AM$ (Encodage). Les points de la matrice Q sont des points de Ep (a, b), qui seront cryptés et envoyés à Brahim.

Brahim déchiffre tous les points cryptés pour obtenir la matrice de retour Q. Une fois Q est généré, Brahim récupère la matrice de message d'origine M à partir de $M = A^{-1}.Q$. En utilisant la matrice M, Brahim récupérera le message d'origine.

6.11 Résultats et comparaison

Nous sommes intéressés par la technique de codage / décodage présentée dans [90], nous voulons la comparer avec notre algorithme modifié de Kolbitz, au moment de l'exécution. On garde l'équation E_{4019} (–10, 16).

Nous commençons à calculer le temps d'encodage et de décodage du massage M = "ⴰⵊⵙⵉ" qui contient 56 = 4 × 14 bits. Nous avons déjà expliqué que chaque caractère de l'alphabet Unicode Tifinagh est codé sur 14 bits (voir paragraphe Tifinagh). Ensuite, nous ajoutons un caractère de l'alphabet Tifinagh au message M pour chaque expérience.

La performance de notre méthode proposée est comparée aux techniques existantes pour prouver son efficacité. Le temps de calcul pour différentes tailles de message M est comparé entre la méthode proposée et les méthodes existantes, comme indiqué dans le tableau 6.8 :

Taille du message M_i en binaire Et ces blocs m_i en décimal	**Valeurs m_{min} , m_{max} ,r_{max} de M_i en décimal**	**Mapping (secs)**		**Decoding (secs)**		**Total time (secs)**	
		Koblitz $\times 10^{-2}$	**[90]**	**Koblitz $\times 10^{-4}$**	**[90]**	**Kioblitz $\times 10^{-2}$**	**[90]**
$M_1 = M$+2D44 = 56 bit M_1={48, 68, 77, 83, 90, 99}	m_{min}=48 m_{max}=99 r_{max}=40	56	2,832	0,3	0,520	86	3,352
$M_2 = M_1$+2D66 = 70 bit M_2={48, 68, 77, 83, 90, 99, 102}	m_{min}=48 m_{max}=102 r_{max}=39	53	3,016	0,4	0,683	93	3,699
$M_3 = M_2$+2D68 = 84 bit M_3={48,68,77,83,90,99,102,104}	m_{min}=48 m_{max}=104 r_{max}=38	43	3,780	0,44	0,831	87	4,611
$M_4 = M_3$+2D69 = 98 bit M_4={48, 68,77, 83, 90,99,102,104, 105}	m_{min}=48 m_{max}=105 r_{max}=37	29	4,348	0,5	0,923	79	5,271
$M_5 = M_4$+2D6E = 112 bit M_5={48, 68,77, 83, 90, 99, 102, 104, 105, 110}	m_{min}=48 m_{max}=110 r_{max}=36	12	4,551	0,53	1,139	65	5,69

Tableau 6.8 : Comparaison de cartographie, encodage / décodage entre la méthode Koblitz et [90] utilisant le message M = ﴿﴾.

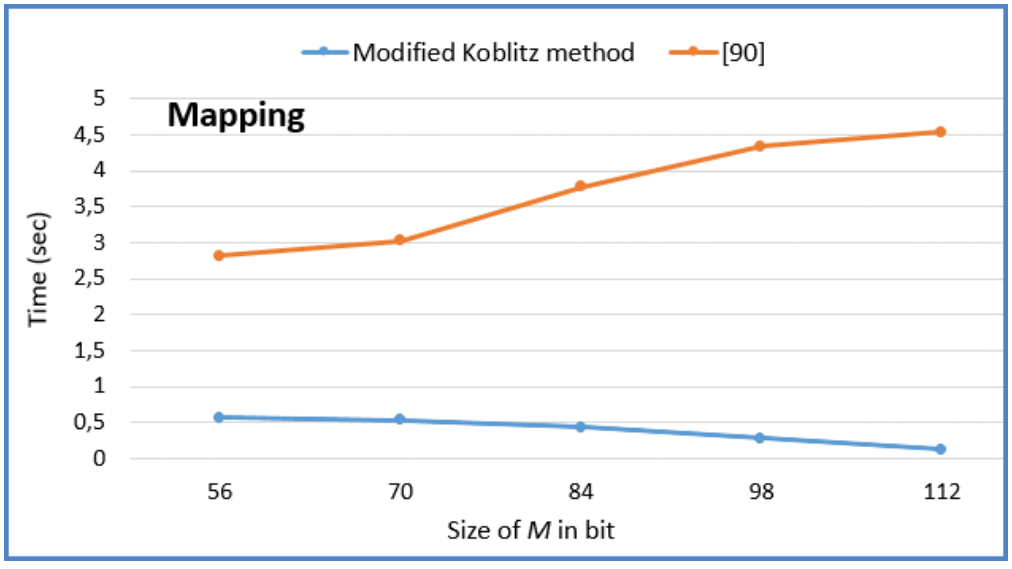

Figure 6.13 : Temps de cartographie de la méthode de Koblitz modifiées et la méthode de la référence [90].

La figure 6.13 montre clairement que le temps de codage de notre méthode proposée est bien inférieur au temps pris par la méthode matricielle de la méthode de la référence [90]. Notre algorithme ne dépend pas nécessairement de la longueur du message M, cela dépend

simplement du nombre d'opérations effectuées entre m_{min} et m_{max}, et du paramètre r_{max}. Notons que le temps de mappage du message M = 112 bits est très petit : 12.10^{-2}secondes par rapport à l'autre taille de M. Notre algorithme a trouvé toutes les valeurs décimales des blocs médians de M = 112 en r_{max} = 36, sans faire d'autres opérations. En revanche, la méthode proposée par [90], dépend de la longueur du message M, il recourt à de nombreux ajouts et doublements des points de la courbe. On également que la méthode [90] utilise les caractères répétés du message M au cours de mappage.

Ceux-ci, explique également la différence du temps de codage/décodage entre les deux méthodes.

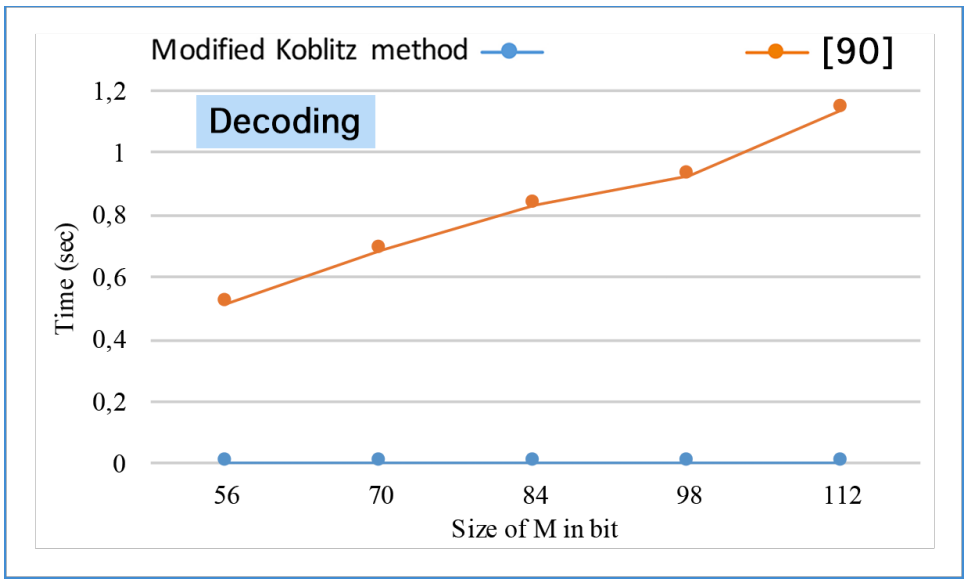

Figure 6.14 : Temps de décodage de la méthode Koblitz modifiée et de la méthode [90]

La figure 6.14 montre également le temps de décodage des deux méthodes; la méthode Koblitz modifié maintient le temps très bas et stable car son décodage ne nécessite que le temps de division $\lfloor x_j/r \rfloor$. Cependant, la méthode de la référence [90] utilise la matrice inverse de codage. Dans la figure 6.15, nous résumons le temps des deux opérations de codage / décodage des deux méthodes

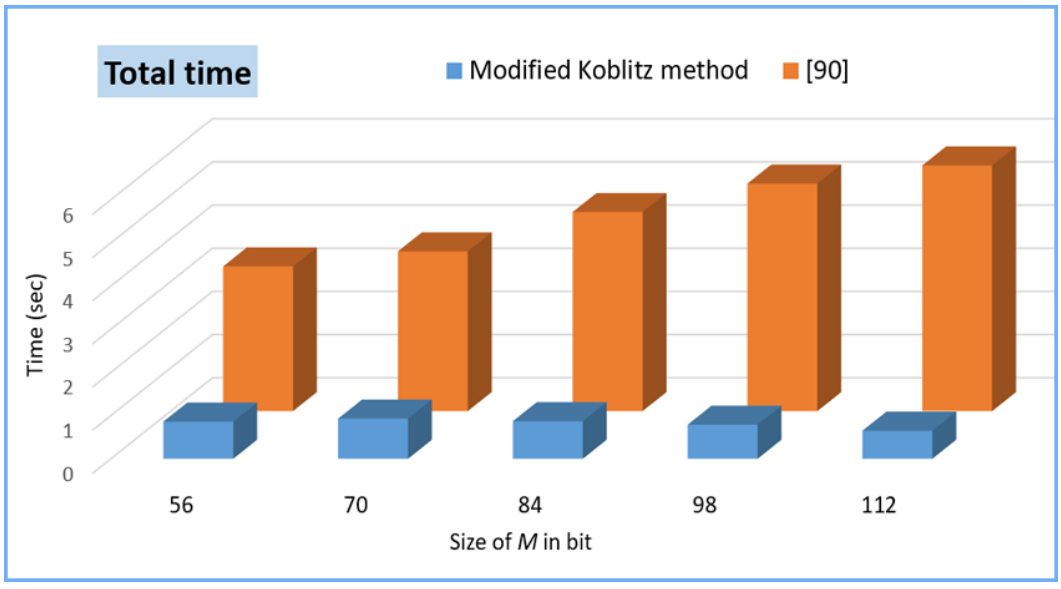

Figure 6.15 : Temps de codage et de décodage de la méthode Koblitz modifiée et [90].

6.12 Conformité de la cartographie probabiliste de Koblitz

Dans cette étude, nous avons essayé de voir le comportement des différentes courbes elliptiques Ep(a, b) choisies en fonction des temps de codage et de décodage d'un message choisi.

Pour cela, nous avons utilisé notre algorithme de Koblitz modifié et un message "ⵣ" à caractère unique pour différents paramètres a, b et p des courbes elliptiques Ep (a, b).

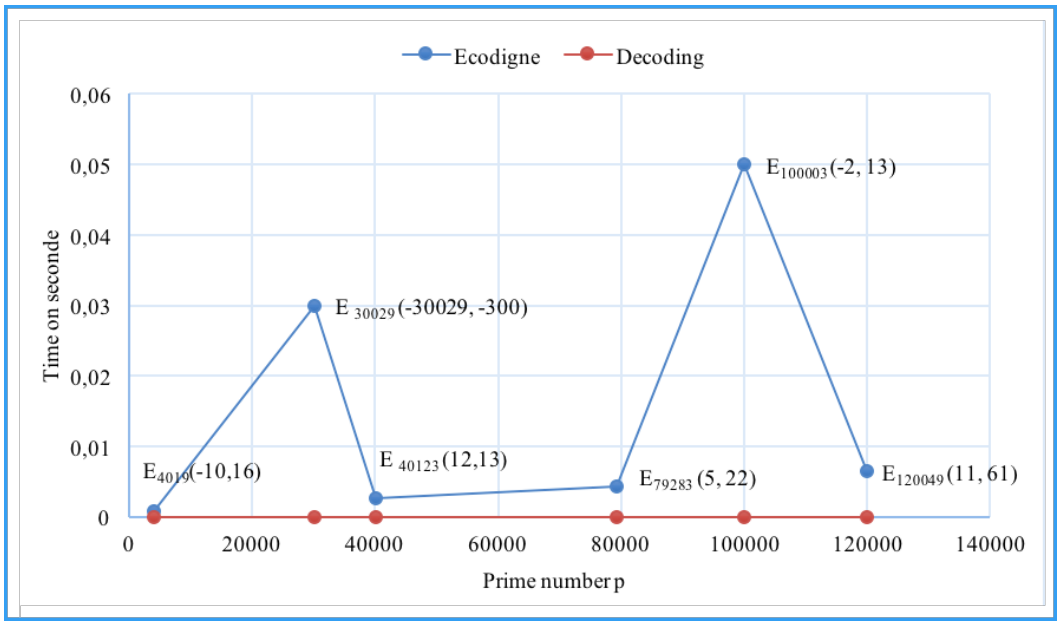

Figure 6.16 : Temps de codage et de décodage pour différentes valeurs : a, b et p de Ep (a, b).

D'après les résultats de la figure 6.16, on remarque que le temps d'exécution du codage est différent par rapport aux différentes valeurs des paramètres de Ep (a, b), alors que le temps d'exécution du décodage est constant et négligeable par rapport à celui du codage. Les variations des temps de codage sont dues à la méthode probabiliste de Koblitz, et les valeurs constantes de décodage sont dues à la division simple de $\lfloor x_j / r \rfloor$.

6.13 Analyse

Pour évaluer plus le comportement de notre méthode proposée avec différentes tailles de texte binaire écrit en alphabet tifinagh, nous avons essayé de calculer le temps de chiffrement et de déchiffrement du texte et l'énergie consommée durant ces deux processus. Avec le changement des tailles des paramètres de la courbe élliptique Ep (a, b).

6.13.1 Méthodologie

H. Durga Tiwari et J.Hyung Kim [91] ont proposé une nouvelle méthode de cartographie du codage ECC par ADN hybride, nous utilisons leurs paramètres de simulation dans notre méthode de Koblitz modifiée.

Ep (a, b) : $y^2 = x^3+537680305x+1059676324 \bmod 3946183951$, point de générateur G(1152222263, 133703258) et son ordre n = 3946206427, la clé privée k = 2454757958, la clé publique P_{pub} = (3539395206, 802765602).

Toutes les expériences ont été conduites sur une machine équipée de processeurs Intel Core i5 de 2,7 GHz et de 8 Go de RAM.

La consommation énergétique est analysée en prenant en compte la consommation électrique

cumulée tout au long du temps d'exécution des algorithmes : d'encodage, de chiffrement, décodage et de déchiffrement.
Ces consommations d'énergie ont été mesurées à l'aide d'Intel Power Gadget 3.5.

6.13.2 Résultats de mise en œuvre et comparaison

Cette fois, nous avons choisi des tailles de texte variant de 84 à 588 bits (multiples de 84 bits), ce qui équivaut à 7 expériences.
Ces tailles de texte sont divisées en 7 blocs de 12 bits, puis sont converties en décimales pour les insérées dans notre algorithme (application) de cartographie proposée.
Les résultats de cartographie, chiffrement et de l'énergie consommée au cours de ce processus sont présentés dans le tableau 19.

Taille de texte (bit)	**Temps de mappage de tous les blocs *Pmi* (ms)**	**Temps de Mapping + chiffrement $P_{Ci} = P_{mi}(x_i, y_i) + kP_b$, (ms)**	**Energie consommé (mj)**
84	0,34548	0,52882	38,28
168	0,33893	0,53146	44,82
252	0,48642	1,52473	129,12
336	1,58642	1,89548	154,48
420	1,66877	2,24785	168,88
504	1,78429	2,44981	189,56
588	2,4688	3,36488	241,63

Figure 6.9 : Déchiffrement, décodage et énergie consommée pour différentes tailles de texte.

Dans la figure 6.17, on note une relation linéaire entre la consommation de temps et la taille du texte saisi pour [91], par contre notre méthode probabiliste est basée sur la division de chaque longueur de message au niveau de la cartographie, ce résultat a été bien expliqué dans la section "**6.12**".

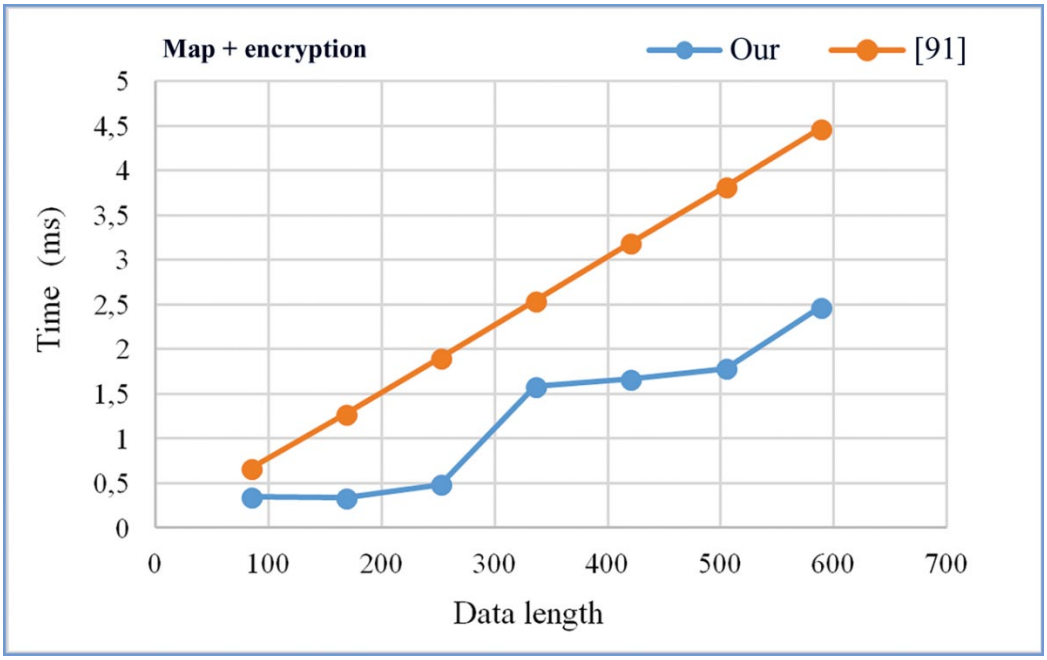

Figure 6.17 : Temps de mappage et de chiffrement pour différentes longueurs de données de $E_{3946183951}$ *(537680305, 1059676324).*

Après le cryptage, le nombre de points Pc_i de chaque longueur de message saisie a été obtenu, tandis que le processus de [91], ont obtenu un message binaire crypté.

Le tableau 20 montre le temps de décryptage, le décodage et la consommation d'énergie de notre méthode proposée

Nombre de points *Pci*pour chaque texte	**Déchiffrement $Pm_i = Pc_i - kP_b$ (ms)**	**Décoding $m_i = \lfloor x_j / r \rfloor$ (ms)**	**Total time (ms)**	**Energy (mj)**
7	0,13321	0,04238	0,17559	13,33
14	0,27251	0,07832	0,35083	27,60
21	0,41832	0,13514	0,57342	42,24
28	0,55168	0,17341	0,72509	57,12
35	0,67311	0,24983	0,92294	72,22
42	0,82179	0,26217	1,08396	89,03
49	1,17891	0,33404	1,51295	102,02

Tableau 6.9 : Déchiffrement, décodage et énergie consommée pour différentes tailles de texte.

La figure 6.18 montre le temps de décryptage et de décodage pris par le schéma de la référence [91] et notre schéma, les chiffres sur les graphiques indiquent le nombre de blocs de 12 bits de message utilisés par notre méthode, tandis que le résultat de la référence [91], indique le nombre de bits à décrypter.

Notez que le nombre de bits augmente pour chaque opération de chiffrement pour le travail de la référence [91]. Par exemple pour notre cas, les 49 blocs entrés pour le chiffrement sont égaux à 49 × 12 = 588 bits, mais dans [91], 24878 bits. Ce qui explique la différence de temps entre les deux méthodes.

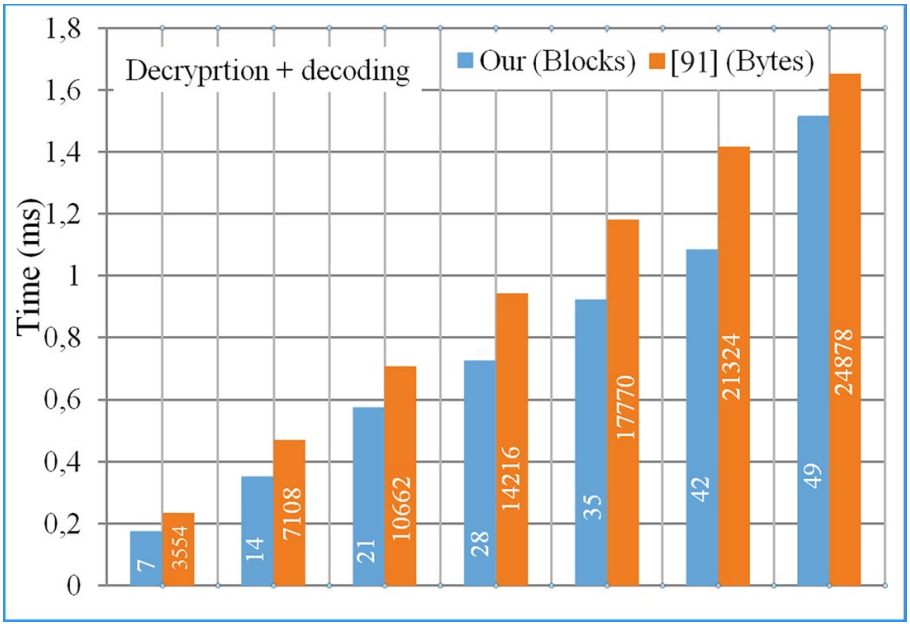

Figure 6.18 : Temps de décryptage + décodage consommé par le schéma [91] et notre schéma.

La figure 6.19 montre la consommation totale d'énergie pendant (cartographie + cryptage) et (décryptage + décodage) de notre approche et celle de [91].

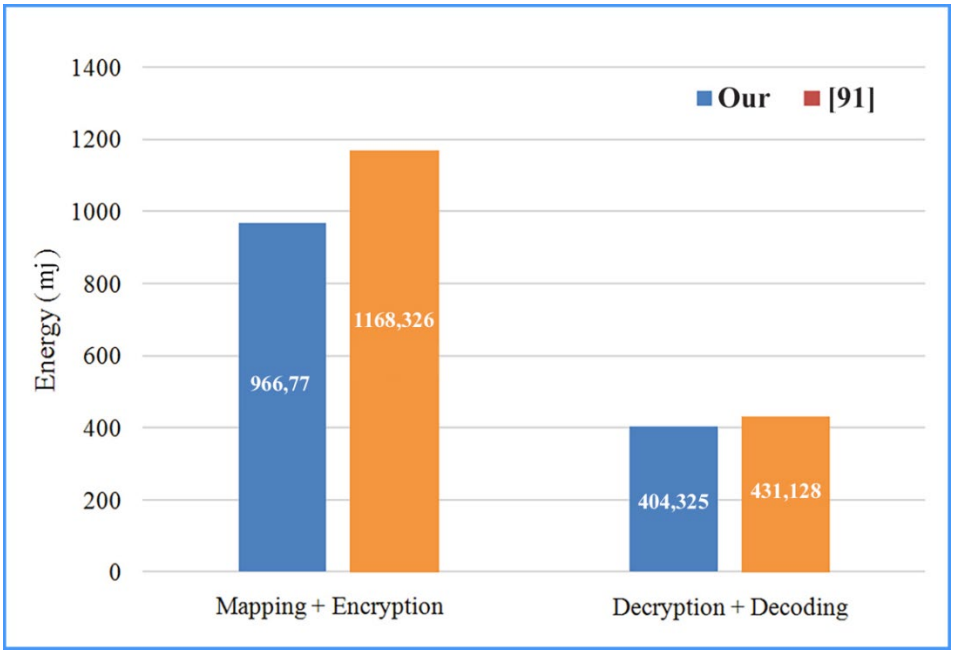

Figure 6.19 : Consommation totale d'énergie pendant (cartographie + cryptages) et (décryptages + décodage) de notre approche et [91].

Il est clair que l'énergie consommée par notre approche est inférieure à l'énergie consommée par celle du travail [91]. Étant donné que leur méthode nécessite un total de 11 étapes pour exécuter leurs processus cryptographiques, celles-ci nécessitent plus de mémoire et de temps car l'aspect de la consommation d'énergie dépend à la fois de la puissance et de la durée d'exécution.

Notez également que l'énergie consommée lors du chiffrement est beaucoup moins importante dans le déchiffrement, car le chiffrement nécessite plus d'opérations d'addition et de multiplication des points de courbe elliptique.

6.14 Conclusions :

Dans ce chapitre, nous avons proposé une amélioration de la méthode de cartographie probabiliste de Koblitz, tout en fournissant deux exemples de mise en œuvre illustrés en détail. Cette méthode réduit le temps de codage, en utilisant les modifications de boucle de l'algorithme de Koblitz initial, et l'élimination des doubles blocs dans les calculs de codage, ainsi qu'une conclusion intéressante sur le temps de codage pour Koblitz qui n'augmente pas nécessairement avec la taille d'un message, car son principe (probabiliste) est de trouver une valeur $1 \leq r \leq r_{max}$ qui permet d'associer tous les blocs d'un message M aux points de la courbe elliptique EC. On note que la seule technique antérieure connue pour mapper des messages non chiffrés aux points EC était probabiliste, afin de donner une faible probabilité d'échec.

La difficulté de la méthode que nous proposons consiste à trouver l'équation Ep (a, b) convenable aux blocs mi de M.

Nous pensons que ce travail contribuera à orienter les futures recherches dans le domaine de la sécurité, car les courbes de Koblitz demandent moins de calculs pour des niveaux de sécurité similaires, ce qui permet des améliorations directes, notamment en termes de temps et de consommation d'énergie.

Cela apparaîtra dans de futurs travaux où nous nous concentrerons sur d'autres types de courbes et de champs elliptiques, notamment les courbes d'Edwards et les courbes de Koblitz sur les champs binaires.

CONCLUSION GÉNÉRALE

onclusion, ce mémoire, par le nombre important de thèmes qu'il aborde et de questions qu'il soulève, nous a donné l'occasion de découvrir plusieurs domaines des mathématiques, assez différents les uns des autres, et nous a sensibilisé à l'importance des mathématiques dans l'étude de nombreux problèmes concrets actuels.

Aujourd'hui la sécurité des réseaux de capteurs est un domaine très important de recherche. Ce besoin accru de sécurité dans les applications de réseaux de capteurs sans fil a suscité nos efforts pour développer des algorithmes et des solutions à ce phénomène.

En général, le problème de la sécurité dans le domaine des réseaux est un problème décisif vu que les données transmises sont évidemment sensibles et il est souvent aisé de les intercepter ou de les manipuler, notamment dans un réseau pourvu d'une composante sans fil, Mais on peut diminuer le degré de danger de différentes attaques précisément qui sont de type déni de service DoS en appliquant des solutions résistantes.

Dans ce contexte, Nous avons introduit les réseaux de capteurs sans fil, leurs utilités, leurs modes de fonctionnement, ainsi que les différents concepts existants dans le domaine de sécurité d'un RCSF. Ensuite, nous avons cité des solutions de défense capables de détecter quelques dangereuses attaques contre les réseaux de capteurs sans fil en se concentrant sur leurs principes de fonctionnement.

L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est celui de routage et le gaspillage de ressource dont l'énergie. Il est donc important de mettre en place des protocoles de routage et d'optimisation d'énergie pour prolonger la durée de vie du réseau. Dans ce sens, nous avons évaluées les performances de plusieurs protocoles d'agrégation de données pour RCSF, nous avons conclu que c'est très difficile d'en choisir un.

La cryptographie sans certificat est une approche efficace étudiée largement dans le chapitre 4 pour deux raisons : premièrement, elle élimine le besoin d'autorité de certification dans l'infrastructure à clé publique et deuxièmement, elle peut résoudre le problème de confiance des clés de la cryptographie basée sur l'identification.

Dans cette partie de notre travail, nous avons proposé un schéma de signature sans certificat basé sur RSA et nous avons prouvé qu'il était sûr sous certaines hypothèses bien étudiées. Nous pensons que le nouveau schéma est plus adapté aux systèmes avec des canaux à faible bande passante et / ou à faible puissance de calcul, ce qui le rend approprié pour RCSF.

La reconnaissance des entités nommées des données des réseaux de capteurs sans fil est une vision passionnante qui permet aux informations structurées d'être interprétées sans ambiguïté. Une interprétation précise est une condition préalable nécessaire à la recherche, à la récupération et au traitement automatiques des données des capteurs.

Le résultat a montré que notre approche pour l'amazighe surmonte d'autres méthodes dans leurs performances et en termes de précision. L'approche hybride atteint 93%, 97%, 74% et 65% pour la précision en Personne, Lieu, Organisation et Divers, respectivement. Notre approche est suffisamment simple pour pouvoir être utilisée pour différentes langues autres que la langue amazighe. Notre méthode peut être appliquée dans tous les domaines d'application notamment les protocoles de routage des réseaux de capteurs sans fils pour minimiser la consommation de l'énergie.

La cryptographie à courbe elliptique n'est pas seulement utilisée pour le chiffrement, mais également pour les protocoles de génération de clé et la signature numérique. ECC utilise moins de mémoire, la génération de paires de clés et la signature sont considérablement plus rapides.

Dans le chapitre 6, nous avons abordé les connaissances de base et les conventions pour comprendre ce qu'est la cryptographie à courbe elliptique, En outre, on démontre comment diviser le texte en clair en blocs pour ECC et comment « mapper » les codes numériques du texte en clair en points de courbe elliptique. Ainsi un algorithme optimisé de multiplication des points d'EC et leur analyse de complexité temporelle sont présentés. Un exemple d'implémentation de l'algorithme de chiffrement/déchiffrement à l'aide de ECC est également illustré en détail.

Par ailleurs, nous avons démontré la fiabilité de notre méthode de cartographie probabiliste de Koblitz, en utilisant un exemple de message écrit en caractères Tifinagh Unicode. Nous avons montré à travers des analyses et des simulations basées sur des comparaisons, que cette méthode réduit le temps et l'énergie de chiffrement durant le processus de chiffrement/déchiffrement. Ainsi la sécurité de la technique de codage qui assurera une double sécurité pour le chiffrement du message transmis.

Nous sommes arrivés à une conclusion intéressante sur le temps de codage de la méthode modifié de Koblitz, qui n'augmente pas nécessairement avec la taille d'un message *M*, car son principe (probabiliste), est de trouver une valeur r ($1 \leq r \leq r_{max}$) qu'elle permet d'associer tous les blocs d'un message *M* aux points de la courbe elliptique EC.

Nous continuions à améliorer les résultats obtenus, ont cherchons un algorithme qui nous permet de trouver une équation cryptographique Ep(a, b) convenable aux blocs *mi* de *M*. pour obtenir de meilleurs résultats possible.

PERSPECTIVES ET FUTURES RECHERCHES

Dans notre première contribution, nous avons proposé une étude de comparaison de huit protocoles d'agrégation de données sécurisées, Nous pensons que notre travail aidera à concevoir les protocoles d'agrégation de données les plus efficaces et plus sécurisés pour les chercheurs. Cependant les RCSF constituent un axe de recherche très fertile et ont de nombreuses perspectives d'application dans des domaines très variés, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans des conditions réelles. En outre, chaque protocole a ses propres contraintes. De ce fait, la conception d'un réseau de capteurs est une tâche très difficile parce qu'elle devra combiner les contraintes propres aux systèmes distribués et aux systèmes embarqués. Pour cette raison, les perspectives ouvertes par ces travaux sont nombreuses et variées. Nous en énumérons quelques-unes :

- La tendance actuelle de la recherche sur l'agrégation de données préservant la confidentialité pour les réseaux statiques. Mais de nombreux scénarios de la vie réelle, tels que la détection de l'état de santé des athlètes et des patients, nécessite des réseaux dynamiques. Par conséquent, l'agrégation de données préservant la confidentialité dans des environnements dynamiques est une future direction de recherche possible.

- De nombreuses décisions critiques sont prises sur la base des résultats agrégés obtenus. Par conséquent, les résultats agrégés doivent être exempts de modifications malveillantes des données de capteur, c'est-à-dire de fausses injections de données, par tout nœud compromis (en particulier par un agrégateur compromis). Détecter les fausses injections de données dans les RCSF est une tâche difficile. Par conséquent, le développement d'un système d'agrégation de données préservant la confidentialité qui évite le problème des modifications malveillantes des données des capteurs sur leur chemin vers l'évier pourrait être un problème intéressant pour les recherches futures.

Dans la deuxième contribution, nous prévoyons évaluer la réponse de notre schéma proposé pour un réseau de capteurs sans fil sur la consommation d'énergie, le temps d'exécution, la ROM et la RAM des capteurs, pour tester son adaptation aux systèmes avec des canaux à faible bande passante et / ou à faible puissance de calcul. Ainsi, nous simulerons notre schéma dans l'outil de simulation de sécurité réseau AVISPA [75]. Nous améliorerons également les performances de notre schéma et nous les comparerons à d'autres schémas de signatures basées sur l'*ID* et sans certificat.

Dans la troisième contribution, les objectifs en perspectives du chapitre 5 est de construire des extracteurs d'entités amazighes de très haute précision pour les noms pour différentes catégories, et leurs dépoilements dans le langage des réseaux de capteurs sans fils, en essayant d'utiliser une approche d'agrégation simple pour répondre à des questions telles que : " quel bâtiment a consommé le plus d'énergie hier ? " et " où est l'endroit le plus froid ? ". Dans cette perspective, nous effectuons des recherches sur les algorithmes d'agrégations des capteurs pour choisir le plus adapté à notre approche hybride, en suite nous sélectionnons un système adéquat pour interroger les données du RCSFs. (Comme le langage de requête SPARQL par exemple).

Dans la quatrième contribution nous pensons que ce travail contribuera à orienter les futures recherches dans le domaine de la sécurité à l'aide des courbes de Koblitz sur les champs finis, et son exploitation dans la sécurité des RCSF. Comme perspective, nous essayons d'étudier ce protocole afin de lui procéder un algorithme de chiffrement/déchiffrement léger.

Nous nous concentrerons sur d'autres types de courbes et de champs elliptiques, notamment les courbes d'Edwardset les courbes de Koblitz sur les champs binaires, dont l'objectif est de trouver l'équation Ep (a, b) cryptographique convenable aux blocs m_i de message M.

ANNEXE

Le répertoire officiel de l'alphabet Tifinaghe-IRCAM avec leurs correspondants en arabe et en caractères latins

TIFINAGHE		Correspondance latine	Correspondance arabe	Exemples
ya	ⴰ	a	ا	ⴰⵏⴰⵔ
yab	ⴱ	b	ب	ⴰⴱⴰⵣⵏ
yag	ⴳ	g	ڭ	ⵜⵣⴳⵛⵣ
yagw	ⴳⵯ	g^w	ڭ'	ⴰⵅⴳⴳⵯⴰⵢ
yad	ⴷ	d	د	ⴰⵀⵙⴷ
yaḍ	ⴹ	ḍ	ض	ⴰⴹⴰⵇ
yey	ⵢ	ə		ⴰⵀⵇⵟⵙⵟⵟⵙ
yaf	ⴼ	f	ف	ⴰⵀⵙⵙ
yak	ⴽ	k	ك	ⵜⵣⵙⴽⵔⵙ
yak^w	ⴽⵯ	k^w	ك'	ⴰⵛⵏⵏⴰⴽⴽⵯⵀ
yah	ⵀ	h	ه	ⴰⵀⵏⵏⵙⵍ
yaḥ	ⵃ	ḥ	ح	ⴰⵃⵍⵓⵄ
yaɛ	ⵄ	ɛ	ع	ⴰⵄⵙⵍ
yax	ⵅ	x	خ	ⵜⵣⵅⵙⵣ
yaq	ⵇ	q	ق	ⴰⵇⵙⵙ
yi	ⵉ	i	ي	ⵉⵛⵉ
yaj	ⵊ	j	ج	ⴰⵄⵍⵔ
yal	ⵍ	l	ل	ⴰⵛⵍⴰⵍ
yam	ⵎ	m	م	ⴰⵛⴰⵍ
yan	ⵏ	n	ن	ⵉⵙⵏⵍ
yu	ⵓ	u	و	ⵙⵏⵛ
yar	ⵔ	r	ر	ⵙⵙⴰⵙ
yaṛ	ⵕ	ṛ	ڕ	ⵙⵕⵕⴰ
yaɣ	ⵖ	ɣ	غ	ⴰⵄⵙⵙⵛ
yas	ⵙ	s	س	ⵉⵍⵙ
yaṣ	ⵚ	ṣ	ص	ⵜⴰⵛⵚⵕⵇⵉⵜ
yac	ⵛ	c	ش	ⴰⵛⵍⵍⴰⵍ

yat	ⵜ	t	ت	ⵜⴰⵅⴰⵙⵜ
yaṭ	ⵟ	ṭ	ط	ⵜⵣⵟⵟ
yaw	ⵡ	w	و^	ⴰⵡⴰⵎ
yay	ⵢ	y	ي^	ⵜⴰⵢⵙⴰ
yaz	ⵣ	z	ز	ⴰⵛⴰⵣⵢ
yaẓ	ⵥ	ẓ	ژ	ⴰⵥⴰⵎⵣⵛ

RÉFÉRENCES

[1] United Nations, Department of Economic and Social Affairs. (2011) World urbanization prospects, the 2011 revision. [Online]. Available : http://esa.un.org/unpd/wup/index.html.

[2] European Telecommunications Standards Institute (ETSI), "Electromagnetic compatibility and Radio spectrum Matters (ERM) ; System Reference document (SRdoc) : Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications,TR 103 055 V1.1.1 (2011-09)," European Telecommunications Standards Institute, Tech. Rep., 2011.

[3] https://fr.wikipedia.org/wiki/Tokyo

[4] A.S.BOUSSAD ,Sécurisation des Réseaux Ad hoc : Systèmes de Confiance et de Détectionde Répliques, ÉCOLE DOCTORALE "Sciences et Ingénierie pour l'Information" Faculté Des Sciences Et Technique ,12-07-2011.

[5] H.GHANNOUM, Etude conjointe antenne/canal pour les communications Ultra Large Bande E n présence du corps humain, École Nationale Supérieure des Télécommunications, 11 Décembre 2006.

[6] M. Hauspie. Spécification et implémentation de la couche de communication sans fil pour Objets Mobiles Communicant. PhD thesis, PhD thesis, Laboratoire d'informatique fondamentale de Lille, 2001.

[7] N. K. Gupta. Inside bluetooth low energy. Artech house, 2016.

[8] S. Farahani. ZigBee Wireless Networks and Transeivers. 1st Edition. Elsevier, 2008.

[9] K. Alagha, G. Pujolle et G. Vivier. Réseaux de mobiles et réseaux sans fil. Eyrolles, 2001.

[10] I. Stojmenovic. Handbook of wireless Networks and Mobile computing. John Wiley & Sons, 2002.

[11] H. Labiod. Etude sur le Wifi pour le conseil stratégique des technologies de l'information(CSTI). ENST, 2002.

[12] F. Siddiqui, S. Zeadally et K. Salah. Gigabit wireless networking with IEEE 802.11 ac: technical overview and challenges. Journal of Networks, vol. 10, no. 3, page 164, 2015.

[13] S. Paolo. Topology Control in Wireless Ad Goc and Sensor Networks. Jhon Wiley & Sons Ltd, 2005.

[14] K. Alagha, G. Pujolle et G. Vivier. Réseaux de mobiles et réseaux sans fil. Eyrolles, 2001.

[15] L. Nuaymi. WiMAX : Technology for BroadbandWireless Networking. John Wiley & Sons Ltd, 2007.

[16] C. Soin, R. Jain et A. Tamimi. Scheduling in IEEE 802.16e mobile WiMAX networks : key issues and a survey. IEEE Journal on selected areas in communications, vol. 27, no. 2, pages 156–171, 2009.

[17] S. Ahmadi. Mobile wimax : A systems approach to understanding ieee 802.16 m radio access technology. AcademicPress, 2010.

[18] A. B. De Aguiar, Neto A. M. S. Cunha R. P. P. et R. F. Pinheiro. A Novel Model for Optimized GSM Network Design. arXiv preprint arXiv :0909.1045, 2009.

[19] C. Lindemann et A. Thummler. Performance analusis of the general packet radio service. In 21st Internationnal Conference on Distributed Computing System 2001.

[20] K. W. Richardson. Umts overview. Electronics & Communication Enfeneering Journal, 2000.

[21] E. Dahlman, S. Parkvall et J. Skold. 4g : Lte/lteadvanced for mobile broadband. Academic press, 2013.

[22] A.S. BOUSSAD ,Sécurisation des Réseaux Ad hoc : Systèmes de Confiance et de Détection de Répliques, ÉCOLE DOCTORALE "Sciences et Ingénierie pour l'Information" Faculté Des Sciences Et Technique ,12-07-2011.

[23] L. Mainetti, L. Patrono, et A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey ", 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), p. 1-6, 2011.

[24] J. Stankovic, "Research Directions for the Internet of Things ", IEEE Internet of Things Journal, vol. 1, no 1, p. 3-9, févr. 2014.

[25] Z. Shelby et C. Bormann, "6LoWPAN: The Wireless Embedded Internet", John Wiley & Sons, 2011.

[26] J. Yick, B. Mukherjee, D. Ghosal, "Wireless sensor network survey", International Journal of Computer and Telecommunications Networking, pp.2292 2330, 2008.

[27] N.LABRAOUI , La sécurité dans les réseau sans fil ad hoc , A L'UNIVERSITE DE TLEMCEN FACULTÉ DES SCIENCES, 2012.

[28] J.M.Rabaey and M.J.Ammer and J.L.da Silva Jr. and D.Patel and S.Roundy , PicoRadio supports ad hoc ultra-low power wireless networking , IEEE Computer Magazine , (2000).

[29] S.E.BENBRAHIM , Défense contre l'attaque d'analyse de trafic dans les réseaux de capteurs sans fil (WSN) , A l'Université De Montréal : École Polyclinique De Montréal , aout 2011.

[30] W.ZNAIDI .Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil 2010.

[31] M.LEHSAINI, Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique ,A l'Université A.B Tlemcen Faculté des Sciences pour l'Ingénieur Université de Franche-Comté U.F.R Sciences et Techniques École Doctorale SPIM ,2011.

[32] N.LAABOULI.La sécurité dans les réseaux sans _l ad hoc.Thése dans Université Abou Bekr Belkaid.2012.

[33] E.HADDAD . Détéction de la retransmission séléctive sur les réseaux de capteurs . Université de Montral .2011.

[34] N.BRAHIM . Routage Multichemin sécurisé pour un réseau capteurs sans fil vidéo " Attaque wormhole : étude et contre mesure". 2012.

[35] D.BOUBICHE. Une approche Inter-Couches (cross-layer) pour la Sécurité dans les R.C.S.F .2013.

[35] http://wapiti.telecomlille1.eu/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2004/Moutaib-Elor/Solutions%20standards/Untitled-2.html

[37] M.DOUIRI .Réseaux de capteurs sans fil : routage et sécurité .2009-2010

[38] Q.Monnet. Modèles et mécanismes pour la protection contre les attaques par déni de service dans les réseaux de capteurs sans fil. 2015.

[39] B.BENSABER .Introduction sur les réseaux de capteurs sans fil. 2012.

[40] S. K. Jain et K. Garg, « A Hybrid Model of Defense Techniques against Base Station Jamming Attack in Wireless Sensor Networks », in First International Conference on Computational

Intelligence, Communication Systems and Networks. 2009.

[41] I.MANSOUR.Contribution à la sécurité des communications des réseaux de capteurs sans fil.Thése dans UNIVERSITÉ BLAISE PASCAL _ CLERMONT II.2013.

[42] http ://fr.wikipedia.org/wiki/Received_Signal_Strength_Indication.

[43] Jing Deng, Richard Han, and Shivakant Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. Security and Privacy for Emerging Areas in Communications Networks, International Conference on, 0 :113–126, 2005.

[44] http ://fr.wikipedia.org/wiki/Attaque_par_analyse_du_trafic.

[45] Ying Jian, Shigang Chen, Zhan Zhang, and Liang Zhang. A novel scheme for protecting receiver's location privacy in wireless sensor networks. Wireless Communications, IEEE Transactions on, 7(10) :3769 –3779, october 2008.

[46] Xi Luo, Xu Ji, and Myong-Soon Park. Location privacy against traffic analysis attacks in wireless sensor networks. In Information Science and Applications (ICISA), 2010 International Conference on, pages 1 –6, april 2010.

[47] Karl, H. and Willig, A. (2005) 'Protocols and architectures for wireless sensor networks', *In Wiley.*

[48] Diallo, C. (2010) 'Techniques d'amélioration du routage et de la formation des clusters multisauts dans les réseaux de capteurs sans fil', *In PhD dissertation, Télécom & Management SudParis, Evry, France*.

[49] S.Ozdemir and Y.Xiao, Secure data aggregation in wireless sensor networks: A Comprehensive overview, Computer Networks 53 , 2022-2037,2009 .

[50] Y.Sang and H.Shen, Secure Data Aggregation in Wireless Sensor Networks:ASurvey.

[51] Yang Y, Wang X, Zhu S, and Cao G. (2008) SDAP: A Secure Hop-by- Hop Data Aggregation Protocol for Sensor Networks, ACM T INFORM SYST SE, 11, pp 1-43.

[52] Yoon, M.; Jang, M; Kim, H.; Chang, J. (2010): A new sensitive data aggregation scheme for protecting integrity in wireless sensor network. *IEEE Computer Society International Conference on Computer and Information Technology*, pp. 2463-2470.

[53] Kim Y, Lee H, Yoon M, et al. (2013) Hilbert-Curve Based Data Aggregation Scheme to Enforce Data Privacy and Data Integrity for Wireless Sensor Networks. NT J DISTRIB SENS N, pp 1-14.

[54] S. Ozdemir, Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism, in: Proceedings of the ICPS'07: IEEE International Conference on Pervasive Services, Istanbul, Turkey, 2007, pp. 165–168.

[55] Girao, J., Westhoff, D. and Schneider, M. (2009) 'Cda : Concealed data aggregation for reverse multicast traffic in wireless sensor networks', *In Proceedings of IEEE International Conference on Communications, ICC2005, Seoul, Korea*.

[56] Boubiche, D. E.; Boubiche, S.; Homero, T. C.; Pathan, A. K.; Bilami, A. et al. (2016): SDAW: secure data aggregation watermarking-based scheme in homogeneous WSN. *Telecommunication Systems*, vol. 62, no. 2, pp. 277-288.

[57] Omar, R.; Merad, B.; Sidi, M. S.; Mohammed, F. (2015): A novel secure aggregation scheme for wireless sensor network using stateful public key cryptography. *Ad Hoc Networks*, vol. 32, no. C, pp. 98-113.

[58] Groat, M. M.; He, W.; Forrest, S. (2011): KIPDA: k-indistinguishable privacy-preserving data aggregation in wireless senor networks. *IEEE INFOCOM*, pp. 2024-2032.

[59] Zhang, W. S.; Wang, C.; Feng, T. M. (2008): GP2S: generic privacy-preservation solutions

for approximate aggregation of sensor data. *IEEE International Conference on Pervasive Computing and Communications*, pp.179-184.

[60] Ertaul L, Yang J.H, Saldamli G. (2015) Analyzing Homomorphic Encryption Schemes in Securing Wireless Sensor Networks (WSN). IJCSNS, 15: 1-11, 5 : Chandra A, Choksi C, Wadhwani M, et al. (2013) A Survey of the Privacy Homomorphism in Wireless Sensor Networks. ICDCIT, pp 46-50.

[61] Harjito B, Potdar V, Bambang Harjito J.S, et al. (2012) Watermarking Technique for Wireless Sensor Networks: A State of the Art. In: Proceedings ICSKG, Beijing, China, pp 253-256.

[62] Christof Paar Axel Poschmann Thomas Eisenbarth, Sandeep Kumar and Leif Uhsadel. A survey of lightweight cryptography implementations. IEEE Design and Test of Computers, 2007.

[63] N. Ghoualmi-zine. « la cryptographie à la physique ». Conférence EIP, sécurité informatique: tendance et application, INI-Alger juin 2006. PAGES 32-45. 2006.

[64] Samuel Galice. « Modèle de sécurité dynamique pour les réseaux spontanés ». Thèse de doctorat. Institut National des Sciences Appliquées de Lyon. 2007.

[65] G. Yuval : How to Swindle Rabin. Cryptologia, 3(3):187–189, 1997. 1.3.1

[66] A. Kartit « Une nouvelle approche de détection d'intrusions et étude des problèmes liés au déploiement de politiques de sécurité dans les réseaux informatiques ». p 129. 2011

[67] K. M. Martin.Everyday Cryptography : Fundamental Principles RSA and Applications. Oxford University Press, 1 mars 2012.

[68] K. McCurley, "Discrete logarithm problem," in Proceedings of Symposia Applied Mathematics , pp. 49-74, 1990.

[69] A. W. Dent. A survey of certificateless encryption schemes and security models. International Journal of Information Security, 7(5):349–377, 2008.

[70] D.Fiore ,R. Gennaro, and N. P. Smart. Relations between the security models for certificateless encryption and ID-based key agreement. International Journal of Information Security, 11(1):1–22, 2012.

[71] Y. H. Hwang, J. K. Liu, and S. S. M. Chow. Certificateless public key encryption secure against malicious KGC attacks in the standard model. Journal of Universal Computer Science, 14(3):463–480, 2008.

[72] G. Yang and C. H. Tan. Certificateless public key encryption: A new generic construction and two pairing-free schemes. Theoretical Computer Science, 412(8–10):662–674, 2011.

[73] Zhang, J., Mao, J.: An efficient rsa-based certificateless signature scheme. Journal of Systems and Software 85(3), 638–642 (2012).

[74] Bellare, Mihir; Rogaway, Phillip (1993). "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". ACM Conference on Computer and Communications Security: 62–73.

[75] Glouche, Y., Genet, T., Houssay, E.: SPAN A Security Protocol Animator for AVISPA. IRISA, September 2008.

[76] J. C. Choon and J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," in Proceedings of the International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography, vol. 2567, pp. 18–30, 2003.

[77] Xun Yi, "An identity-based signature scheme from the Weil pairing," IEEE Communications Letters, vol. 7, no. 2, pp. 76–78, 2003.

[78] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in Advances

in Cryptology-ASIACRYPT, vol. 2894 of Lecture Notes in Computer Science, pp. 452–473,
Springer, 2003.

[79] M. C. Gorantla and A. Saxena, “An efficient certificateless signature scheme,” in Proceeings of the International Conference on Computational Intelligence and Security, vol. 3802, pp. 110–116, 2005.

[80] Sullivan, N. A “Primer on Elliptic Curve Cryptography”. Available online: https://arstechnica.com/security/2013/10/a-relatively-easyto-understand-primer-on elliptic curve-cryptography/ (accessed on 24 October 2013).

[81] N.Koblitz, “Elliptic Curve Cryptosystems,” Mathematics of Computation, vol. 48, no. 177, pp. 203-209, 1987.

[82] V. S. Miller, "Use of Elliptic Curves in Cryptography," Advances in Cryptology - CRYPTO '85: Proceedings, pp. 417-426: Springer-Verlag, 1986.

[83] Jeffrey Hoffstein, Jill Catherine Pipher, Joseph H Silverman, and Joseph H Silverman. An * introduction to mathematical cryptography, volume 1. Springer, 2008.

[84] Neal Koblitz. Algebraic aspects of cryptography, volume 3. Springer Science & Business Media, 2012.

[85] R. Schoof, “Counting Points on Elliptic Curves over Finite Fields”. Journal de Theorie des Nombres. de Bordeaux 7, 219-254, 1995.

[86] http://mathworld.wolfram.com/EllipticCurve.html

[87] J. Lopez, R. Dahab (2000), “An overview of elliptic curve cryptography”, Technical report, IC- 00-10, May 22. Available at http:// www.dcc.unicamp.br/ic-main/public- cation - e.html

[88] S. VANSTONE,D.A. Next generation security for wireless : elliptic curve cryptography, http://www.compseconline.com/hottopics/hottopic20_8/Next.pdf.

[89] J. WALTER The role of ECDSA in wireless communication (implementation and evaluation of ECDSA on constrained devices), Los Angeles 2002.

[90] F. Amounas, B. Noble, and I. N. Sneddon, “Fast Mapping Method based on Matrix Approach For Elliptic Curve Cryptography” International journal of information & Network security, vol. 1, No.2, June 2012, pp. 54-59.

[91] H.Durga Tiwari and J.Hyung Kim, “Novel Method for DNA-Based Elliptic Curve Cryptography for IoT Devices”. ETRI Journal, Volume 40, Number 3, June 2018.

[92] L.Zenkouar, L’écriture amazighe tifinaghe et Unicode, Étude et Documents Berbères, 2004, pp.175-173.

[93] Renaud, « Le développement des objets connectés : les nouveaux chiffres de 2018 », OBJETCONNECTE.NET, 24-janv-2017.

[94] https://www.challenge.ma/maroc-masque-intelligent-de-detection-du-coronavirus 137480/

[95] « Automotive Sensors and Electronics 2017 -- Conference and Expo ». [En ligne]. Disponible sur: http://www.automotivesensors2017.com/. [Consulté le: 19-sept-2018].

[96] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott AVanstone. Handbook of applied cryptography . CRC press, 1996.

[97] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners. Springer Science & Business Media, 2009.

[98] https://fr.wikipedia.org/wiki/%C3%89tats-Unis .

[99] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli, "Design of a low-latency, high-reliability wireless communication system for control applications," in Proc. IEEE Int. Conf. Commun. (ICC), 2014, pp. 3829 – 3835.

[100] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli, "Design of a low-latency, high-reliability wireless communication system for control applications," in Proc. IEEE Int. Conf. Commun. (ICC), 2014, pp. 3829 – 3835.

[101] Tadilo Endeshaw Bogale, Xianbin Wang and Long Bao Le Western, Machine Intelligence Techniques for Next-Generation Context-Aware Wireless Networks, pp : 2, 12 jan 2018.

[102] Thangaramya K, Kulothugan K, Logambigai R, Selvi M, Ganapathy S, Kannan A (2019) Energy aware cluster and nero-fuzzy based routing algorithm for wireless sensor networks in IoT.

[103] Sethukkarasi R, et al (2014) An intelligent neuro fuzzy temporal knowledge representation model for mining temporal patterns, pp 1167–1178. Comput Netw 151:211–223.

[104] Richa Sharma · Sudha Morwal · Basant Agarwal · Ramesh Chandra · Mohammad S. Khan, A deep neural network-based model for named entity recognition for Hindi language, pp 1-2, Springer 04 April 2020.

[105] F. Gramegna, S. Ieva, G. Loseto, and A. Pinto, "Semantic-enhanced resource discovery for coap-based sensor networks," in Proceedings of the 5th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI), June 2013.

[106] C. Perera, A. Zaslavsky, P. Christen, M. Compton, and D. Georgakopoulos, "Context-aware sensor search, selection and ranking model for internet of things middleware," in Proceedings of the 14th International Conference on Mobile Data Management (MDM), vol. 1, June 2013, pp. 314–322.

[107] M. Molina, J. Sanchez-Soriano, and O. Corcho, "Using open geographic data to generate natural language descriptions for hydrological sensor networks," Sensors, vol. 15, no. 7, pp. 16 009–16 026, 2015

[108] K.-K. Yap, V. Srinivasan, and M. Motani, "Max: human-centric search of the physical world," in Proceedings of the 3rd international conference on Embedded networked sensor systems. ACM, 2005, pp. 166–179.

[109] M. Molina, J. Sanchez-Soriano, and O. Corcho, "Using open geographic data to generate natural language descriptions for hydrological sensor networks," Sensors, vol. 15, no. 7, pp. 16 009–16 026, 2015.

[110] R. G. Raskin and M. J. Pan, "Knowledge representation in the semantic web for earth and environmental terminology (sweet)," Computers & geosciences, vol. 31, no. 9, pp. 1119–1125, 2005.

[111] Syed Ali Imran Quadri, P.Sathish, "IoT Ba sed Home Automation and Surveillance System", International Conference on Intelligent Computing and Control Systems(ICICCS)(IEEE), 2017.

[112] Ameur, M., Bouhjar, A., Boukhris, F., Boukouss, A., Boumalk, A., Elmedlaoui, M., Iazzi, M. et Souifi, H. (2004). Initiation à la langue Amazighe. Rabat, Maroc: IRCAM.

[113] M.Outahajala, Apprentissage d'un étiqueteur morphosyntaxique de la langue amazighe, 06 juin 2015 : p 18.

[114] John D. La_erty, Andrew McCallum, and Fernando C. N. Pereira. Conditional randomelds: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 282{289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[115] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional

random _elds, feature induction and webenhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, pages 188{191, Morristown, NJ, USA, 2003. Association for ComputationalLinguistics.

[116] Yassine Benajiba, Mona Diab, and Paolo Rosso. Arabic named entity recognition using optimized feature sets. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 284{293, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[117] Georgi Georgiev, Preslav Nakov, Kuzman Ganchev, Petya Osenova, and Kiril Simov. Feature-rich named entity recognition for bulgarian using conditional random _elds. In Proceedings of the International Conference RANLP-2009, pages 113{117, Borovets, Bulgaria, September 2009. Association for Computational Linguistics.

[118] F. Erik and Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei, Taiwan, pp. 155-158.

[119] Nirenburg, S.; McShane, M.; Beale, S.; Wood, P.; Scassellati, B.; Magnin, O.; Roncone, A. Toward Human Like Robot Learning. In Proc. NLDB 2018, to appear; 2018, Paris, France.

[120] J.-H. Kim, H. Kwon, D.-H. Kim, H.-Y. Kwak, and S.-J. Lee. "Building a serviceoriented ontology for wireless sensor networks". Computer and Information Science, pp. 649–654, 2008.

[121] F. ataa allah, S. Boulaknadel. " La promotion de l'amazighe à travers les technologie de l'information et de la communication, Asinag, 9, 2014, p. 33-48.

DOCTORAT

AVIS DE SOUTENANCE DE THÈSE

Le Doyen de la Faculté des Sciences de Rabat annonce que :

Karim EL MARSSI

soutiendra une thèse intitulée :

La sécurité des réseaux de capteurs sans fil et leur application pour la langue amazighe.

En vue de l'obtention du **DOCTORAT**

Discipline : **Science de l'ingénieur.**

Spécialité : **Informatique et Télécommunications.**

Centre de recherche : **Rabat IT Center.**

Responsable de centre de recherche : **Pr. Bouchaïb BOUNABAT.**

Structure de recherche **: Laboratoire de Recherche en Informatique et Télécommunications.**

Responsable de structure de recherche : **Pr. Mohamed OUADOU.**

Directeur de Thèse : **Pr. Mohamed EL MARRAKI.**

Co-Directeur de Thèse : **Pr. Ali Kartit.**

Date : **25/12/2020.**
Heure : **15h00.**
Lieu : **Amphi ALHAYTAM.**

Résumé

Les réseaux de capteurs sans fil ont gagné un intérêt majeur lors de la dernière décennie. Cependant, la sécurité reste un problème fondamentalement ouvert.
Notre mémoire s'articule autour de quatre contributions :
Le chapitre 3 examine les protocoles d'agrégation sécurisée de données proposés, à la pointe de la technologie. Ces protocoles peuvent être classés et analysés lors de l'agrégation de données chiffrées de saut par saut, bout en bout, et l'agrégation sécurisée de données non chiffrées. Dans le but d'évaluer l'efficacité des protocoles mentionnés.
Le chapitre 4 propose un schéma de signature sans certificat basé sur RSA, qui est un schéma largement appliqué dans des scénarios réels, en particulier pour les réseaux de capteurs sans fil. D'après les résultats, ce schéma est sécurisé dans le modèle d'oracles aléatoires et que sa sécurité est étroitement liée au problème de logarithme discret de RSA. En effet, nous avons prouvé qu'il était sûr sous certaines conditions contre les adversaires A_1 et A_2, ainsi il garantit également l'intégrité, la non répudiation et l'authentification.
Le chapitre 5 traite la reconnaissance des entités nommées REN des données des réseaux de capteurs sans fil. Cette approche permet aux informations structurées d'être interprétées sans ambiguïté. Une interprétation précise est une condition préalable nécessaire à la recherche, à la récupération et au traitement automatique des données des capteurs. Notre recherche sur RE a été appliqué sur la langue amazighe.
Le chapitre 6 propose une amélioration de la méthode de cartographie probabiliste de Koblitz en utilisant un message écrit en caractères Tifinaghe Unicode. L'amélioration proposée porte sur les variables de codage de l'algorithme de Koblitz pour faciliter la recherche des racines carrées de l'équation CE correspondant aux blocs du message M. Les résultats expérimentaux ont montré que notre algorithme de cartographie Koblitz modifié nécessite beaucoup moins de temps de codage qui ne varie pas nécessairement avec la taille du message M.

Mots-clefs : Réseaux de capteurs sans fil, Sécurité, Cryptographie, Entités Nommés de la langue amazighe, Courbe elliptique, protocoles de routage, Signature sans certificat.

Abstract

Wireless sensor networks have gained major interest over the past decade. However, security remains a fundamentally open issue.

Our work revolves around four contributions:
Chapter 3 examines the proposed state-of-the-art secure data aggregation protocols. These protocols can be classified and analyzed during hop-by-hop, end-to-end, encrypted data aggregation and secure aggregation of unencrypted data. In order to evaluate the effectiveness of the mentioned protocols.
Chapter 4 provides an RSA-based certificateless signing scheme, which is a scheme widely applied in real-world scenarios, especially for wireless sensor networks. According to the results, this scheme is secure in the random oracle model and that its security is closely related to the discrete logarithm problem of RSA. Indeed, we have proven that it is safe under certain conditions against opponents A_1 and A_2, thus it also guarantees integrity, non-repudiation and authentication.
Chapter 5 deals with the recognition of entities named REN from data from wireless sensor networks. This approach allows structured information to be interpreted unambiguously. Accurate interpretation is a prerequisite for finding, retrieving and automatically processing sensor data. Our research on RE has been applied to the Amazigh language.
Chapter 6 proposes an improvement of the Koblitz probabilistic mapping method using a message written in Tifinaghe Unicode characters. The proposed improvement relates to the coding variables of the Koblitz algorithm to facilitate the search for the square roots of the CE equation corresponding to the blocks of the message M. The experimental results have shown that our modified Koblitz mapping algorithm requires much less encoding time which does not necessarily vary with the size of the M.

Keywords: Wireless sensor networks, Security, Named Entities of the Amazigh language, Elliptic Curve, Cryptography, Routing protocols, Signature certificateless.