Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

En Vue de l'Obtention du Diplôme de Master en Informatique

Option : Réseaux et Systèmes Distribués

Thème :

Etude sur l'Applicabilité de la Cryptographie Asymétrique aux Réseaux de Capteurs sans Fil

Réalisé par :

M^{lle} ATTAF Nassima & M^{lle} CHERFA Hamida

Devant le jury composé de :

Président :	M^{r} AMROUN	Kamel	, M.A.A.
Examinateur :	M^{r} TOUAZI	Djoudi	, M.A.A.
Examinateur :	M^{r} DEMOUCHE	Mouloud	, M.A.B.
Encadreur :	D^{r} OMAR	Mawloud	, M.C.B.
Co-Encadreur :	M^{r} SAADI	Mustapha	, M.A.A.

Juin 2012

Remerciements

Nous rendons grâce à notre Dieu, le tout puissant et miséricordieux, pour nous avoir donné le courage et la patience pour mener à bout ce modeste travail.

Nos plus sincères remerciements s'adressent à notre encadreur D^r OMAR Mawloud maître de conférences et chef de département informatique de l'université de Béjaïa, pour nous avoir proposé ce sujet intéressant et pour ses précieux conseils et encouragements, sans lesquels cette étude n'aurait pas vu le jour. Merci pour votre confiance, votre disponibilité et le temps que vous avez bien voulu consentir à l'aboutissement de ce mémoire.

Nos remerciements s'adressent également à notre co-promoteur M^r SAADI Mustapha, et aux membres de jury, en l'occurrence M^r AMROUN Kamel, M^r TOUAZI Djoudi et M^r DEMOUCHE Mouloud d'avoir accepté d'évaluer notre travail et pour l'intérêt qu'ils y portent.

Nous tenons à remercier tous les enseignants, qui ont assuré notre formation durant notre cycle universitaire, particulièrement D^r TARI Abdelkamel.

Nos remerciements vivement le cher cousin M^r SAIDANI Boualeme et sans oublier M^r ALLOUT Hachemi et M^{r}MADI Kemal, qui ont pu nous assister durant la préparation de ce mémoire.

Enfin un grand merci à nos familles, pour leur soutien permanent et indéfectible qui nous ont permis de chercher au plus profond fond de nous même la force, la volonté et la persévérance à même d'arriver à cet instant des plus important de notre vie et à nos amis et tous ceux qui ont contribué de près ou de loin à la concrétisation de cette oeuvre.

Dédicaces

Je dédie ce modeste travail :

à mes parents, à mes grands parents,
à mon frère Mohand et à ma sœur Katia,
à mes oncles, à mes tantes, à tous mes cousins :
Salwa, Yanis, Fawzi, Kossayla, Lydia, Lamia, Thilleli, Aris
Rahaf, Salima, Kassa, Hanan, Nassima, Nassim, Nora, Jad, Ilina
à toute ma famille de prés et de loin , à mes collègues, à tous
les étudiants de master2 informatique(promotion2012), à tous mes amis,
à Dahia, Tiziri, Sonia, Kahina, Maya, Alaa, et je pense tous particulièrement
à mes deux meilleures amies : Dalal Toudji & Nassima Attaf, ainsi à leurs familles.
Hamida

***Je dédie ce modeste travail** :*

à mes parents, à mes grands parents
à mon frère Samir et sa femme Tassadite
à mes deux nièces Alyssia et Chanez
à mon frère Nabil et sa femme Sabrina
à mon frère Sofiane à mes sœurs Thinhinane et Sélia
à mes oncles et mes tantes à mes cousins et cousines
à toute ma famille de prés et de loin à mes amis à mes collègues,
à tous les étudiants de master2 informatique (promotion 2012),
à tous mes amis : Thiziri, Kahina, Dahia, je pense particuliérement
à mes meilleures amies : Hamida Cherfa, Dalal Toudji, Ratiba Belkassemi,et leurs familles.
Nassima

Que la paix d'Allah soit avec tous... !

Table des matières

Liste des Abréviations

AES	Advanced Encryption Standard.
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
CA	Certifcation Authorit
CPU	Central Processing Unit.
CRL	Certifcation Revocation List.
DES	Data Encryption Standard.
ECC	Elliptic Curve Cryptography
ECDH	Elliptic curve Diffie-Hellman.
ECDSA	Elliptic Curve Digital Signature Algorithm.
GM	Goldwaser Micali.
GPS	Girault - Poupard/Paillès - Stern.
GQ	Guillou Quisquater.
LLL	Lenstra Lenstra Lovasz.
MD5	Message Digest5.
NIST	National Institute of Standards.
NSS	NTRU Signature Scheme.
NTRU	Number Theorists Research Units.
PGCD	Plus Grand Commun Diviseur.
PKI	Public Key Infrastructure.
QS	Crible Quadratique.
RA	Register Authority.
RC4	Rivest Cipher 4.
RCSF	Réseau de Capteurs Sans Fil.
RSA	Rivest Shamir Adleman
SHA	Secure hach Algorithme.
SVP	Shortest Vector Problem
WSN	Wireless Sensor Network.

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

LE développement actuel de technologie offre de nouvelles perspectives dans le domaine des télécommunications sans fil. Cela a rendu possible l'essor de capteurs multifonctionnels avec des coûts réduits, qui sont dotés d'une capacité à communiquer par diffusion radio à portée réduite. Ces capteurs sont des dispositifs de taille réduite avec des ressources très limitées, capables de traiter des informations et de les transmettre à une autre entité sur une distance limitée à quelques mètres. Les réseaux de capteurs sans fil (ou RCSF) utilisent un grand nombre de ces capteurs pour établir un réseau sans infrastructure. Un capteur analyse et surveille une zone définie, et des domaines aussi variés que l'industrie, la recherche environnementale ou la médecine, et propage les données récoltées aux capteurs appartenant à sa zone de couverture. Chaque capteur relayant l'information sur sa propre zone de couverture, rend le réseau entièrement couvert. Par ailleurs, l'énergie limitée des capteurs et les environnements hostiles dans lesquels ils pourraient être déployés, sont des contraintes qui rendent les RCSF très vulnérables. De même que, l'absence de sécurité physique et la nature vulnérable des communications radio sont des caractéristiques qui augmentent les risques d'attaques contre ce type de réseaux.

La confidentialité, l'authenticité et l'intégrité des échanges sont des services de sécurité indispensables, pour certaines applications des RCSF, notamment lorsqu'il s'agit de transporter des informations qui peuvent révéler un secret ou des informations sensibles quand il s'agit de prévenir des accidents catastrophiques comme dans les réacteurs nucléaires, leur sécurité est requise pour mener à bien leurs opérations. La cryptographie, est l'un des mécanismes qui permettent de garantir ces services, car elle représente une science de chiffrement de données, en faisant appel à des clés de cryptage. La gestion de clés est un service très important pour assurer les services de sécurité, et il est considéré comme l'un des aspects les plus difficiles de la configuration. Il fournit des mécanismes efficaces, sécurisés et stables, tel que la cryptographie à clés symétriques (aussi appelée cryptographie à clés publiques) est une solution qui fournit des mécanismes de sécurité très robustes. Cependant, elle exige un espace mémoire assez grand et de haute capacité de calcul, ce qui rend sa mise en œuvre une problématique pour les RCSF. Provenance de ses contraintes, la conception d'un système de gestion de clés représente un sérieux défi.

Le travail effectué, dans le cadre de ce mémoire, est une étude approfondie, que nous l'estimons complète et exhaustive, des systèmes de chiffrement asymétrique afin de ressortir ceux les plus adaptés à la nature et aux contraintes des RCSF. Cette étude est faite selon des critères particuliers, à savoir la complexité de calcul, taille des clés, rapidité de génération des clés, ainsi que les services de sécurité que peut le système assurer.

Ce mémoire est organisé en cinq chapitres. Dans le premier chapitre, nous donnons une brève description sur les RCSF en termes d'architecture, caractéristiques et domaines d'application. Dans le deuxième chapitre, nous donnons quelques notions de base liées à la cryptographie ; des concepts importants pour la suite du mémoire. Dans le troisième chapitre, nous donnons une vue globale sur les différentes attaques, ainsi que les solutions proposées pour le cadre des RCSF. Le quatrième chapitre englobe la première partie de notre contribution qui comporte la taxonomie que nous avons proposée et l'étude des différents systèmes de chiffrement à clés publiques en termes d'avantages, limites et complexité. Le cinquième chapitre comporte la deuxième partie de notre contribution, dans laquelle nous comparons techniquement l'ensemble des systèmes de chiffrement à clés publiques en termes de configuration matérielle requise pour le cadre des RCSF. Enfin, nous clôturons le mémoire avec une conclusion générale.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

Les avancées et techniques dans le domaine des réseaux sans fil, de la micro-fabrication et des microprocesseurs intégrés, ont donné naissance à une nouvelle génération de réseaux à dimension réduite, avec un coût raisonnable, qui sont dotés de capacité à communiquer par diffusion radio à portée réduite : il s'agit des réseaux de capteurs sans fil.

1.1 Les réseaux de capteurs sans fil (RCSF)

Un capteur est un petit appareil autonome doté d'une batterie, il est capable d'effectuer de simples mesures sur son environnement, tels que la température, les vibrations et la pression [1]. Un nœud capteur est constitué des composantes principales suivantes (cf. figure 1.1) [2, 3, 8, 115] :

1. **L'unité d'acquisition** : elle permet d'obtenir des mesures numériques sur les paramètres environnementaux, elle convertit l'information relevée et la transmet à l'unité de traitement.
2. **L'unité de traitement** : elle est composée d'une interface pour l'unité d'acquisition et une autre pour l'unité de transmission. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission.
3. **L'unité de communication** : elle est responsable de toutes les émissions et réceptions de données via un support de communication radio entre l'émetteur et le récepteur.
4. **L'unité d'énergie** : elle est généralement ni rechargeable ni remplaçable. La capacité d'énergie est limitée au niveau des capteurs, cela représente la contrainte principale lors de la conception de protocoles pour les RCSF.

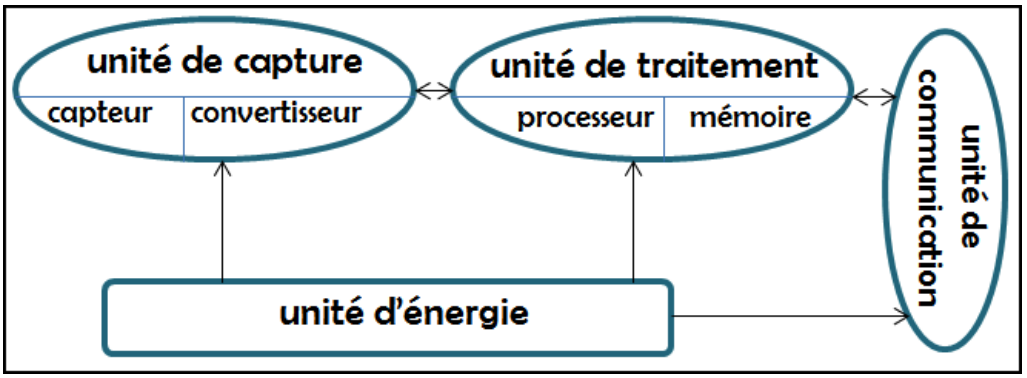

FIGURE 1.1 – L'architecture d'un capteur

Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseaux ad hoc. Il est composé d'un nombre de nœuds déployés à l'intérieur d'une zone d'intérêt afin de la surveiller. Ces nœuds ont pour fonctions de capturer, mémoriser, traiter et communiquer les données vers une station de base [3, 4, 5, 7]. Les nœuds capteurs sont dispersés dans une zone d'intérêt, chacun de ces nœuds a la possibilité de collecter les données et de les router vers une ou plusieurs stations de base(station de traitement). Cette dernière transmet les données collectées à l'utilisateur final à travers un réseau de communication [3, 5]. L'architecture d'un RCSF est illustrée sur la (figure 1.2).

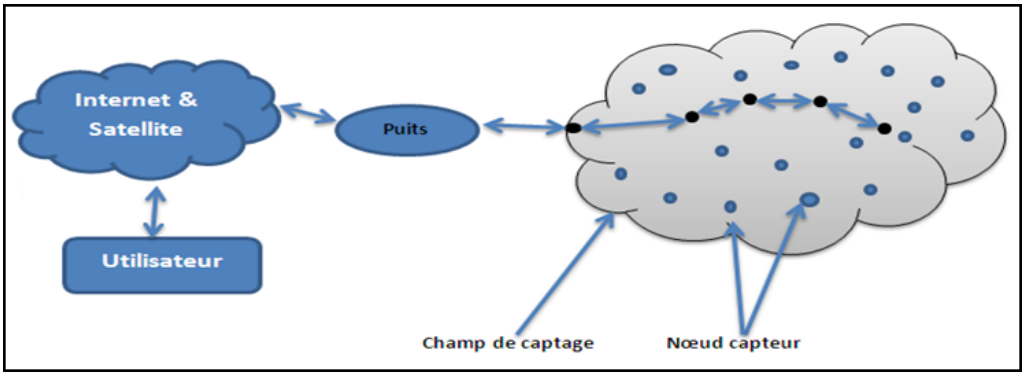

FIGURE 1.2 – L'architecture d'un RCSF

Les RCSF sont caractérisés par [3, 8] :

- Le nombre de nœuds est plus important que le nombre des nœuds dans un réseau ad hoc.
- La densité du réseau est beaucoup plus importante.
- Les nœuds capteurs sont très limités en termes d'énergie, capacité de traitement, capacité de stockage et transmission.
- Les RCSF sont plus touchés par le paramètre de sécurité que les réseaux filaires, cela se justifie par le manque de défense contre les interférences ou le bruit de communication sans fil.
- Les nœuds sont déployés dans un environnement sans infrastructure, n'ayant aucune information sur la topologie du réseau. Pour cela, les nœuds doivent établir l'infrastructure de communication durant la phase d'initialisation, qui doit leurs permettre de répondre aux requêtes venant des nœuds distants, et d'interagir avec l'environnement physique ainsi de transmettre

les données captées via une communication multi sauts.

1.2 Domaines d'application des RCSF

Les applications des réseaux de capteurs sans fil couvrent des domaines divers et variés. Nous pouvons citer des applications militaires, surveillance environnementale, médicale, domestique, commerciale [2, 5] :

- **Applications militaires** : un réseau de capteurs peut être déployé dans un endroit hostile, afin de surveiller :
  - Les mouvements des forces ennemies.
  - Analyser un terrain avant d'envoyer des troupes.
  - Détection des armes chimiques, biologiques ou radiations.
- **Applications de surveillance** : l'intégration des capteurs dans de grandes structures tels que les ponts ou les bâtiments aidera à :
  - Découverte de catastrophes naturelles tels que : détection d'incendie, détection d'inondation, séisme, etc.
  - Déceler les altérations dans la structure suite à un séisme ou au vieillissement.
- **Applications environnementales**
  - La détection de débuts d'incendies par des thermos capteurs.
  - Le contrôle de la qualité de l'air par des capteurs chimiques dans les milieux urbains.
  - Le contrôle des sites industriels vis-à-vis des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole).
- **Applications médicales** : dans le domaine de la médecine, des microcapteurs qui pourront être avalés ou implantés sous la peau pour assurer la télésurveillance des organes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées. Cela permet le suivi et le contrôle des patients dans des hôpitaux.
- **Applications domestiques** : avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que :
  - Les fours à micro-ondes.
  - La climatisation.
- **Applications commerciales** : l'application des nœuds de capteurs dans le domaine commercial permet le :
  - Contrôle environnemental dans les usines et les bureaux.
  - Contrôle d'inventaire.

1.3 Les facteurs et contraintes de conception

Il existe plusieurs facteurs de conception des RCSF, nous énumérons [1, 2, 3, 5, 6, 132] :

- **Tolérance aux pannes :** c'est la capacité de maintenir le fonctionnement des RCSF sans interruption due à une erreur intervenue sur un ou plusieurs capteurs.
- **Scalabilité :** le nombre de nœuds capteurs disséminés dans une surface donnée doit être dans l'ordre de centaines ou selon l'application à réaliser. Par conséquent le réseau doit être capable de fonctionner avec un grand nombre de capteurs tout en permettant son évolution.
- **Contraintes matérielles :** la principale contrainte est liée aux faibles dimensions d'un nœud capteur, telle que la taille exigée peut être plus petite qu'un centimètre cube. Ceci nécessite l'intégration des circuits électroniques à haut niveau.

1.4 Sécurité

Les RCSF connaissent actuellement une grande extension et une large utilisation dans différents types d'applications, dont celles exigeant une grande sécurité. Les capteurs disposent des capacités mémoire et de stockage ainsi de calcul très limitées, ces contraintes font que l'application des mesures classiques de sécurité est restreinte. Les réseaux de capteurs sans fil connaissent certaines vulnérabilités [3, 132] :

- L'écoute ou perturbation des messages échangés.
- Les nœuds eux-mêmes sont des points de vulnérabilité du réseau, car une attaque peut compromettre un composant laissé sans surveillance.
- L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion pour l'accès aux ressources.
- Les mécanismes de routage sont d'autant plus critiques dans les RCSF, chaque nœud participe à l'acheminement des paquets à travers le réseau, de plus les messages de routage transitent sur les ondes radio.

Les contraintes de sécurité sont différentes, cependant, dans la plupart d'entre elles, l'intégrité et l'authenticité des données doivent être fournies pour s'assurer que des nœuds non autorisés ne peuvent pas injecter des données dans le réseau. Le chiffrement des données est souvent requis pour des applications sensibles telles que les applications militaires ou les applications médicales [3].

1.5 Conclusion

Dans ce chapitre nous avons présenté des généralités sur les réseaux de capteurs sans fil, tel que nous avons motionné les principaux concepts liés à leur architecture, caractéristiques, et domaines d'applications, ainsi les services utilisés.

CHAPITRE 2

NOTIONS ELÉMENTAIRES SUR LA CRYPTOGRAPHIE

Pendant de nombreuses années, la problématique de la cryptographie a été totalement dérobée comme un art réservé aux militaires et aux espions, c'est au cours de ces dernières années avec le développement des moyens de communication qu'elle est devenue une science fondée sur des techniques mathématiques et informatiques. Le but primordial de la cryptographie est de garantir des fonctionnalités essentielles qui sont des objectifs de la sécurité tels que l'authentification, la confidentialité, et l'intégrité.

2.1 La cryptographie

Le mot 'cryptographie' vient de mots grec `cryptos` 'cachet' et le verbe `graphien` 'écrire' qui signifie étude des écritures secrètes [3]. Donc la cryptographie est définie comme une science des codes secrets et l'art de créer des cryptogrammes, de telle façon que ces derniers sont compris uniquement par leur destinataire légitime [21, 70].

En fonction du nombre de clés utilisées, nous distinguons deux familles de cryptographie, la cryptographie symétrique nécessite que les systèmes de chiffrement et de déchiffrement disposent de la même clé, tandis que la cryptographie asymétrique ou à clés publiques considère deux clés complémentaires une clé publique et autre privée réalisant l'une le chiffrement et l'autre le déchiffrement.

2.1.1 La cryptographie symétrique

Elle utilise une même clé secrète pour chiffrer et déchiffrer des données, dont elle assure la confidentialité. 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 entité communicante dans un groupe de n entités est difficile et conduit à un grand nombre de clés à gérer : ($\frac{1}{2}$*n*(n-1)).
Tel que ce résultat est trouvé a partir de graphe (2.1), comme suit ; le nœud *A* partage des clés avec (n-1) nœuds voisins et le nœud *B* partage des clés avec n-2, ainsi de suite, donc on aura [*(n-1)+(n-2)+...+0*] qui est une somme d'une suite arithmétique.

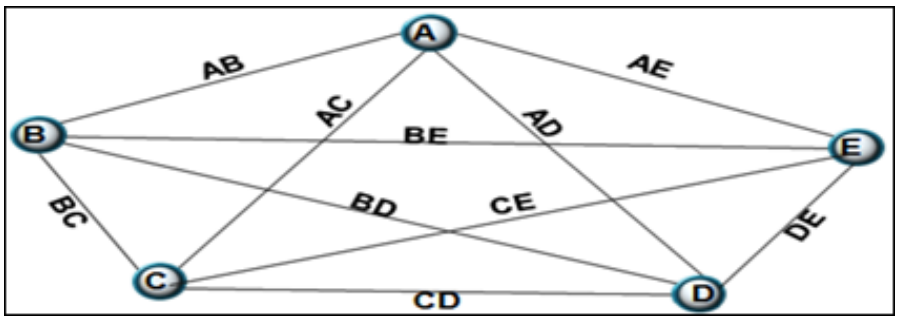

FIGURE 2.1 – Graphe des clés de cryptographie symétrique

Les algorithmes de chiffrement symétrique sont décomposés en [64] :

- Le chiffrement par flux : il se fait bit à bit sans attendre la réception entière des données, l'algorithme le plus connu est le RC4 [131].
- Le chiffrement par bloc : consiste à diviser les données en blocs de taille fixe (64,128), chaque bloc ensuite sera chiffré [48]. Les algorithmes les plus connus sont : DES [130], 3DES [102, 133], et AES [64, 119].

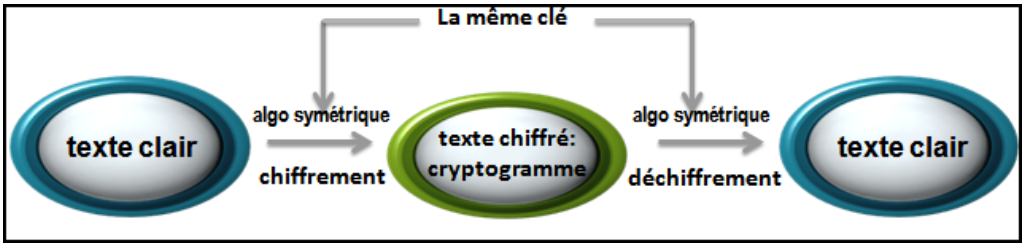

FIGURE 2.2 – Le principe de cryptographie symétrique

2.1.2 La cryptographie asymétrique

Le principe de ce cryptosystème est basé sur l'usage d'un couple de clés, l'une publique qui est connue par tout le monde et l'autre privée qui doit être confidentielle. Les algorithmes asymétriques

les plus connus sont : RSA [43], El Gamal [65], Merkle-Hellman [39], Rabin [42] et ECC [74]. Le fonctionnement de la cryptographie asymétrique est illustré sur la figure 2.3, tel que :

1. L'entité A génère une paire de clés ; l'une publique notée P(A), l'autre privée notée S(A).
2. L'entité B désire communiquer avec A, donc elle récupère la clé publique de P(A).
3. B chiffre le message avec la clé publique P(A), et envoie le message chiffré à A.
4. A déchiffre le message reçu en utilisant sa clé privée P(S).

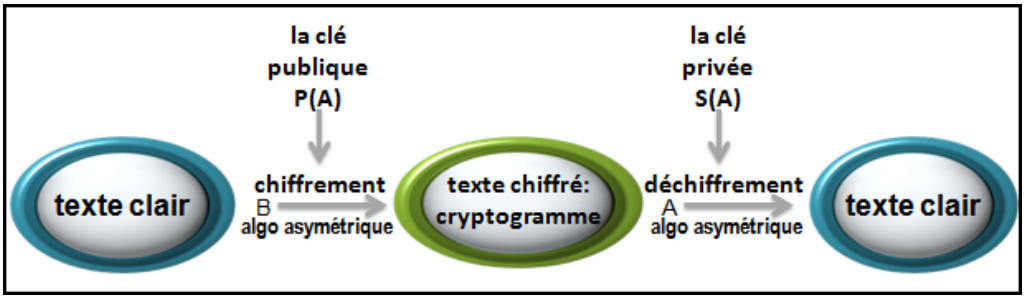

FIGURE 2.3 – Le principe de la cryptographie asymétrique

2.1.3 La fonction de hachage

C'est une fonction mathématique à sens unique qui permet de chiffrer un message dont son déchiffrement est impossible, dont l'objectif est de fournir un résultat représentatif du contenu d'un message de taille restreinte (fonction de condensation) à la taille initiale et ainsi de garantir l'intégrité de ce dernier [3, 44]. Les propriétés de ces fonctions de hachage sont les suivantes [4] :

- La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair (cela signifie que la moindre modification du document entraîne la modification de son haché).
- Un résultat sur un nombre limité d'octets.
- L'impossibilité de retrouver le message original à partir de condensé (sens unique).

Les algorithmes de hachage les plus répandus sont : MD5 [134] (Message Digest) et SHA [127].

2.2 L'authentification des correspondants

2.2.1 Les certificats

Une entité pour prouver que la clé publique lui appartient, elle fera recours à un certificat, qui représente un document numérique contenant toutes les coordonnées utiles d'un utilisateur. Le certificat est signé pour authentifier l'origine du certificat ainsi que son intégrité [16]. Les principaux paramètres d'un certificat sont :

- Version du certificat.
- Numéro de série.
- Signature du certificat.
- Algorithme de chiffrement utilisé pour signer le certificat.
- Fonction de hachage.
- Nom de l'organisme qui génère le certificat.
- Période de validité.
- Identité de l'utilisateur du certificat.
- Clé publique de l'utilisateur.

2.2.2 La signature numérique

Contrairement au certificat, la signature répond à la question comment être sûr de l'émetteur ? Elle consiste à appliquer une fonction de hachage sur une portion du message [37, 54, 77, 86]. Un schéma de signature numérique est composé de :

- La fonction de signature est paramétrée par une clé secrète propre au signataire, elle associe à tout message clair une signature.
- La fonction de vérification permet à partir du message clair et de la signature de vérifier l'authenticité de ce dernier.

Un schéma de signature doit donc posséder un certain nombre de propriétés, en particulier, il doit être en pratique impossible de contrefaire une signature [66] :

- Seul le propriétaire de la clé secrète peut signer en son nom.
- La signature ne doit plus être valide si le message clair est modifié.
- Il doit être impossible de réutiliser une signature.
- Le signataire ne doit pas pouvoir nier la signature d'un message.

Le schéma de chiffrement asymétrique en faisant appel à la signature numérique est illustré dans la figure 2.4 : Tels que :

1. Une entité A signe le message m avec sa clé privée,

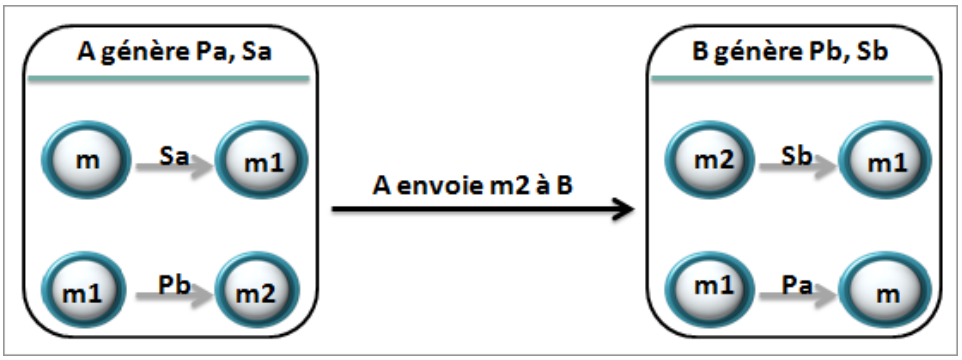

FIGURE 2.4 – Le schéma de singature numérique

2. Puis elle chiffre le résultat m_1 avec la clé publique de l'entité B, et envoie m_2 à B,
3. L'entité B à son niveau va déchiffrer le message (m_2) avec sa clé sucrète,
4. Puis elle va déchiffrer le résultat avec la clé publique de l'entité A, et trouve finalement le message m que l'entité A avait envoyé.

Seulement avec la clé publique de l'entité A que l'entité B peux déchiffrer le message obtenu, cela garantie l'*authentification.* Et aussi le message ne peut pas être altéré, donc l'*intégrité* est assuré, et la *non-répudiation*.
Notamment ce processus de signature, alourdi encore plus le processus de chiffrement asymétrique qui est déjà un processus lent. Une solution qui permet de garantir ces services tous en évitant d'augmenter le temps de calcul et ainsi de réduire la taille des messages, consiste à utiliser une fonction de *hachage*.

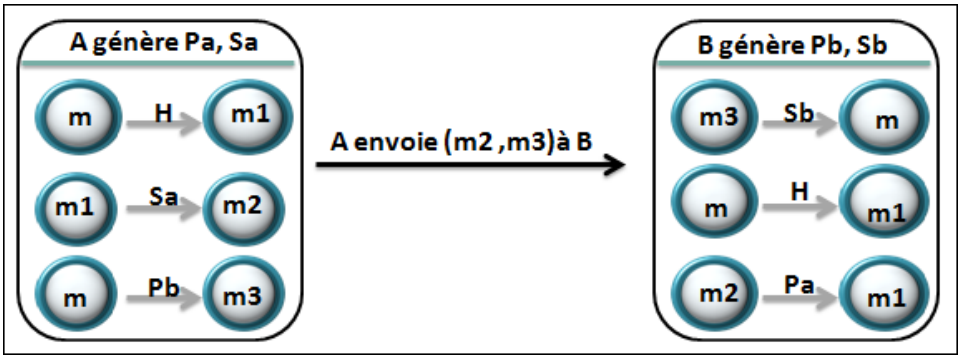

FIGURE 2.5 – Le chiffrement asymétrique avec la fonction de hachage

La figure 2.5 illustre le fonctionnement de chiffrement asymétrique en utilisant la fonction de hachage. Soit m le message que l'entité A veut envoyer à l'entité B :

1. A calcule le haché de message m : H(m)=m1
2. A signe le haché avec sa clé privée : Sa(m1)=m2
3. A crypte le message m avec la clé publique de l'entité B : Pb(m)=m3
4. A envoie le message chiffré m3 ainsi sa signature m2

5. B déchiffre le message m3 avec sa clé privé : Sb(m3)=m
6. B calcule le haché de message obtenu a son niveau : H(m)=m4
7. B vérifie la signature de haché envoyé par A : Pa(m2)=m5
8. Il faut que m4=m5 (le haché calculé=le message signé aprés le décryptage)

2.3 PKI (*Public Key Infrastructure*)

C'est un ensemble d'organisations et procédures qui permettent l'implémentation et la distribution des certificats du cryptographie à clés publiques, il est constitué des éléments suivants [9, 45] :

- Autorité de certification (CA) : est un organisme de confiance, ou un ensemble de ressources (logicielles et matérielles) et de personnels définis par son nom et sa clé publique qui a pour rôle la création , la signature et la publication des certificats ainsi la révocation.
- Autorité d'enregistrement (RA) : c'est l'intermédiaire entre le détenteur de la clé et CA. Il vérifie les requêtes des utilisateurs et les transmet à CA [17].
  Les clés publiques de CA sont sauvegardées dans des navigateurs et ne nécessitent pas l'installation de la part des usagers.

2.4 Système de gestion des clés publique

La gestion des clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Ce mécanisme doit permettre de générer des clés et de les distribuer à leurs utilisateurs d'une manière sécurisés, et ainsi de pouvoir les enregistrer et les gérer de manière sûre [15].

FIGURE 2.6 – Le système de gestion des clés publique

Une entité fait confiance à une autre entité, si cette dernière se comporte comme la première le prévoit. De cette définition, nous pouvons tirer trois cas de relations de confiance [16] :

- Si A fait confiance à B, B n'a pas besoin de faire confiance à A.
- Si A fait confiance à B, B fait confiance à A.
- Si A fait confiance à B, et B fait confiance à C, alors A et B peuvent faire confiance à C.

Le dernier cas est le plus répandu dans les architectures de sécurité, où la troisième entité C est une tierce partie de confiance, c'est une entité spéciale, elle représente l'autorité de certification dans les infrastructures PKI dans les cryptosystèmes à clés publiques.

2.5 Notions mathématiques utilisées dans la cryptographie asymétrique

1. **Diffie-Hellman**
   Diffie-Hellman est une méthode spécifique d'échange de clés qui permet à deux entités de partager une clé k [19, 116] :
   – L'entité A génère aléatoirement a, calcule $g^a \ mod \ p$ et transmet cette valeur à B.
   – L'entité B génère aléatoirement b, calcule $g^b \ mod \ p$ et la transmet à A.
   – La clé k= g^{ab} mod p.
   Tels que (a,b,g et p sont des entiers).

2. **L'équation diophantienne**

   Elle s'écrit sous forme $ax+by = c$ tels que a, b, et c, sont des entiers connus. x et y sont des entiers inconnues. Sa résolution s'appuie sur l'algorithme d'Euclide, le théorème de Bézout [136].

3. **Algorithme d'Euclide étendue**

   Cet algorithme permet, à partir de deux entiers a et b, de calculer leurs couples de coefficients de Bézout ; deux entiers u et v tels que a.u + b.v= PGCD(a, b), tel que a et b sont premiers entre eux et u est l'inverse de a par rapport à b. Cet algorithme détemine quand une équation diophantienne ax+by = c possède une solution, et calcule la solution [105].

4. **Théorème de Bézout**

   Le théorème de Bachet Bézout est un résultat d'arithmétique élémentaire, qui prouve l'existence de solutions à l'équation diophantienne linéaire : d'inconnues x et y entiers relatifs [116]. Ce théorème affirme que l'équation admet des solutions si et seulement si les entiers relatifs (a,b) sont premiers entre eux.

5. **L'algorithme des restes chinois**

   C'est un résultat d'arithmétique modulaire traitant de résolution de systèmes de congruences. Ce théorème est utilisé en théorie des nombres [116]. Pour trouver x = $y^{\frac{1}{2}}$ mod n, tel que n est un produit de deux nombres premiers p et q qui sont congrus à 3 modulo 4. Dans ceci la résolution de l'équation mène à résoudre le système d'équations :
   x_1=$y^{\frac{1}{2}}$ mod p =x $^{\frac{p+1}{4}}$ mod p
   x_2 = $y^{\frac{1}{2}}$ mod q = x $^{\frac{q+1}{4}}$ mod q
   Et ces deux résultats sont des racines de y tels que ${x_1}^2$ mod n=y et ${x_2}^2$ mod n=y.

6. **L'équation de Weirrstrass**

   C'est une forme simplifiée de l'équation d'une courbe elliptique. La simplification de la forme générale à la forme de Weierstrass peut se faire par changement de variable, mais dépend de la caractéristique du corps commutatif K sur lequel la courbe elliptique est définie [50].

7. **Théorème de Fermat-Euler**

   La fonction indicatrice d'Euler, notée ϕ(n), c'est le nombre d'entiers naturels strictement positifs premiers avec n est strictement inférieurs à n. si p est premier, ϕ(p) = p -1 [106].

8. **Homomorphisme de groupe**

   C'est une application entre deux groupes qui respecte leurs structures. Plus précisément, si (G,*)et (G',*) sont deux groupes d'éléments neutres respectifs e et e', une application F : G $\rightarrow$ G' est un homomorphisme du groupe lorsque [116] : $\forall$(x,y)$\in$ G2 , F(x,y)= F(x)* F(y).

9. **Suite supercroissante**

   Chaque élément de cette suite est strictement supérieur à la somme de tous les éléments qui le précède [39, 106].

10. **Problème du Logarithme discret**

    Etant donné g, p et A, tel que A=g^x mod p, il est difficile de retrouver x. Ce problème sera réduit dans l'ordre de son difficulté en appliquant l'algorithme squar and multiple [23].

11. **La distance de Haming**

    C'est un concept qui peut jouer un rôle important dans la théorie des codes correcteurs, elle permet de quantifier la différence entre séquences de symboles [139]. Exemple : a=(0001111), (b=1101011) donc la distance d=3.

12. **Un algorithme glouton**

    C'est un algorithme qui suit le principe de faire étape par étape un choix optimum local [140].
    Soit la suite super-croissante S = (2, 5, 8,17, 35) et s = 42.
    35 est le plus grand élément de S < s, 35 doit intervenir dans le calcul de s.
    Le solde vaut 7 ; 5 étant le plus grand élément de S inférieur à 7, il doit, selon le même raisonnement, intervenir dans le calcul de s. Le nouveau solde étant 2, le problème est résolu : 42 = 2 + 5 + 35.

13. **L'algorithme LLL**

    Il est introduit par A.Lenstra, H.Lenstra et L. Lovász, il permet la réduction de réseau.
    Il prend en entrée un nombre *d* de vecteurs de base d'un réseau, tels que ces vecteurs sont de dimension *n* et de norme inférieure à la base B, et retourne en sortie une base de réseau LLL-réduite, en temps polynomial [26, 96].

14. **Les codes correcteurs**

    c'est une technique de codage basé sur la redondance, son rôle est de corriger les erreurs lors d'une transmission d'une information sur une voie peu fiable, qui peut être altérer d'un moment à un autre [111].

15. **Les codes de Goppa**

    Sont des codes correcteurs, qui construit au début un polynôme formel à partir des symboles à transmettre et de les sur échantillonner), et dans ce cas le résultat est envoyé, car la redondance de ce sur échantillonnage permet au récepteur du message encodé de reconstruire le polynôme même s'il ya eu des erreurs pendant leur transmission [126].

16. **Attaque exhaustive**

    C'est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles. Cette méthode de recherche exhaustive ne réussit que dans les cas où le mot de passe cherché est constitué de peu de caractères [24].

17. **Matrice génératrice**

    On appelle matrice génératrice d'un code linéaire C toute matrice dont les lignes forment une base de code C [113]. Exemple :

    G la matrice génératrice : $G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$

    G est la matrice génératrice du code :
    00000, 00110, 01100, 11110, 10111, 10001, 01111, 01001
    La distance de ce code est 2 (nombre de bits changés partant d'un terme à un autre est 2).

18. **La matrice de permutation**

    C'est une matrice carrée qui vérifie les propriétés suivantes [137] :
    – Les termes ϵ {0,1} ;
    – Il y a un et un seul 1 par ligne.

– Il y a un et un seul 1 par colonne.

19. **La matrice inversible (non singulière)**

C'est une matrice carrée A d'ordre n telle qu' il existe une matrice B d'ordre n [138] : AB = BA = In. Où In désigne la matrice unité d'ordre n.

20. **Résidus quadratique**

On dit que x est résidu quadratique modulo n s'il existe y tel que y^2=x mod n. Si p est premier impair, on définit le symbole de légende [116] : (x/p) = 1 si x carré dans $Z\backslash Z_p$ (x/p) = -1 sinon

21. **Le symbole de légende**

Soit p un nombre premier, pour tout entier x, on définit (x/p)de la façon suivante [116] :
– (x/p) = 0 si p devise x.
– (x/p) = 1 si x est un résidu quadratique modulo p.
– (x/p) = -1 si x n'est pas un résidu quadratique modulo p.
Pour le calculer, nous utilisons (x/ p) = $(x)^{(p-1/2)}$

22. **Le symbole de Jacobi**

Est une généralisation du symbole de Legendre, le symbole de Jacobi d'un nombre x est défini comme suit [116] :

- Si n est un produit de deux nombre premiers p et q, et x est un résidu non quadratique modulo n $\rightarrow \frac{x}{n}= \frac{x}{p} * \frac{x}{q}$ = (x^p mod q)∗ (x^q mod p).

- Si n n'est pas premier $\rightarrow$:
  – Théorème fondamental $(\frac{x}{n})$=$(-1)^{\frac{(x-1)}{2}*\frac{(n-1)}{2}}$ $(\frac{n}{x})$.
  – Première loi complémentaire : $(\frac{-1}{p})$=$(-1)^{\frac{p-1}{2}}$
  – Deuxième loi complémentaire : $(\frac{2}{p})$=$(-1)^{\frac{p^2-1}{8}}$

23. **La sécurité sémantique**

Elle est définie à travers un jeu entre l'adversaire et un challengeur, l'adversaire fournit deux messages en clair de même taille au challengeur qui en chiffre un des deux choix au hasard. L'adversaire doit distinguer quel message a été choisi au hasard et a été chiffré [62].

24. **Un groupe**

Un groupe est une structure algébrique. Soit E un ensemble muni d'une loi de composition interne LCI , on dit que (G,*) est un groupe si [116, 23] :

- * est associative
- (G,*) admet un élément neutre(e neutre si e*x=x).
- Tout élélment de G admet un élément symétrique(x*x'=e).

25. **Réseau euclidien**

C'est une grille régulière de points dans Rn, si b =(b1,..., bd) est une famille de vecteurs linéairement indépendants de Rn, le réseau euclidien engendré par b, noté L(b), est l'ensemble des combinaisons linéaires à coefficients entiers de b1,..., bd, telle que b est une base du réseau L(b) [141].

26. **Complexité des opérations mathématiques**

C'est un comportement asymptotique qui permet de mesurer les performances d'un algorithme et de le comparer avec d'autres réalisant les mêmes fonctionnalités [46].
Il existe plusieurs classes [46] :

- *Classe P* : classe des problèmes "faciles", résoluble en un temps polynomial.
- *Classe NP* : classe des problèmes à vérification " facile ", dont la vérification s'effectue en temps polynomial de la taille des mots.
- *Classe NP complet* : classe des problèmes aussi "durs" que chacun des NP.

27. **Tableau des complexités des opérations arithmétiques**

Le tableau 2.1 permet de calculer les différentes complexités [14, 64]

Opération	Compléxité
Add (a+b)	O log(a)+O log(b)=O log(n)
Mult (a*b)	O log(a)*O log(b)=O $\log(n)^2$
Modulo Mult (a*b) mod n	O $\log(n)^2$
Modulo Exp a^k mod n	O $\log(n)^3$
Euclide etendue	$5* k$
a^e mod n	e
a^n	n-1

TABLE 2.1 – Tableau de complexité

2.6 Conclusion

Nous avons abordé dans ce chapitre les différentes notions élémentaires sur la cryptographie, ainsi quelques concepts mathématiques utilisés dans la cryptographie asymétrique. Dans le chapitre suivant, nous allons définir les différents types d'attaques ainsi que les menaces et les vulnérabilités liées aux RCSF.

CHAPITRE 3

LA SÉCURITÉ DANS LES RÉSEAUX DE CAPTEURS SANS FIL

La sécurité informatique est l'ensemble des politiques et des mécanismes de protection et de contrôle mis en œuvre pour réduire les vulnérabilités d'un système contre les menaces pour éviter les erreurs, afin d'assurer le bon fonctionnement de tel système. Les RCSF sont plus vulnérables en termes de sécurité à cause des liens sans fil et les contraintes matérielles qui les caractérisent.

3.1 Les concepts de base

Du point de vue de la sécurité informatique :

- **Une menace :** est une violation potentielle de la sécurité, qui peut-être accidentelle ou intentionnelle.
- **La vulnérabilité :** c'est une faiblesse dans le système qui peut être exploité par une menace.
- **Le risque :** est généralement caractérisé par l'équation : risque=vulnérabilité+menace
- **L'Attaque :** représente les moyens pour exploiter une vulnérabilité, qui peut être passive ou active.
  - Attaque passive : consiste à écouter ou copier des informations sans les modifier.
  - Attaque active : consiste à altérer ou couper des informations.

3.2 Les objectifs de la sécurité dans RCSF

La problématique de la cryptographie est aussi utilisé dans le domaine des réseaux de capteurs sans fil, tel que la plupart des recherches s'appliquent afin d'améliorer les performances. Les principaux objectifs de la sécurité dans les RCSF sont [3, 6, 7, 15] :

- ***L'authentification*** Elle sert à contrôler et identifier les nœuds afin de coopérer au sein des

RCSF sans risque. En effet, ni la confidentialité ni l'intégrité ne peuvent être assurées si l'authentification est mal gérée, en outre un attaquant peut se joindre au réseau et injecter des messages erronés.

- ***L'intégrité*** Elle assure que les données reçues n'ont pas été altérés durant leur transit dans le réseau. Elle peut être assurée par des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique.
- ***La confidentialité*** La confidentialité consiste à préserver le secret des messages échangés et ne pas les révéler aux adversaires, seulement les entités autorisées peuvent lire le message.
- ***Non repudiation*** vise à empêcher un utilisateur de nier l'envoie ou la réception d'une information, en prévenant qu'un échange ait eu lieu.
- ***La disponibilité :*** Elle signifie que le réseau est disponible pour assurer ses services aux parties communicantes lorsque ceci est nécessaire.
- ***La fraîcheur*** Ce service permet de garantir que les données échangées sur le réseau sont actuelles et ne sont pas une réinjection de précédents échanges interceptées par un attaquant.
- ***Contrôle d'accés*** ce service consiste à limiter l'accès à des personnes privilégiées, et empêcher un accès au tout élément étranger du système

3.3 Les obstacles et problèmes de sécurité dans les RCSF

La sécurité dans les RCSFS est une préoccupation de tous les instants, elle doit être développée de manière à réduire les risques. Notament, quelles sont les obstacles et les problèmes liés à la sécurité des RCSF [3, 4, 6, 7, 32] ?

1. **Des ressources limitées**
   L'utilisation des algorithmes de sécurité nécessitent des ressources en énergie et en calcul, ainsi en mémoires pour la mémorisation des données et les clés utilisées.
   - **Limitation en énergie** : les batteries des capteurs sont ni rechargeables ni remplaçables, cela implique que la ressource d'énergie emportée avec les capteurs doit être conservée pour allonger leur vie et celle du réseau entier.
   - **Mémoire et éspace de stockage limités** : le capteur a un espace mémoire et de stockage limité, ainsi une faible vitesse de calcul. De ce fait, les codes de sécurité et les données relatives telle que les clés de chiffrement doivent être de petites taille.
2. **Communication non fiable**
   Les données sont transmises dans l'air, chaque capteur qui se trouve dans le rayon de couverture peut écouter les messages échangés et l'application d'un bruit sur le canal peut rendre les capteurs incapables de transmettre les messages vu que le média peut apparaître comme occupé en permanence.

3. **Les risques inattendus**
   Puisque les réseaux de capteurs se placent habituellement dans des environnements hostiles, dont leur déploiement les rendent très sensibles à des attaques physiques qui détruisent les capteurs de façon permanente, contrairement à d'autres type d'attaques, de sorte que les pertes sont irrévocables.

3.4 Les attaques dans les RCSF

Les attaques possibles dans les réseaux de capteurs sans fils sont [3, 4, 6, 7, 32] :

1. **Ecoute passive du réseau :** si les données ne sont pas chiffrées, l'attaquant qui dispose d'un équipement puissant(ressource en énergie et en mémoire et vitesse de calcul), peut collecter les informations échangées dans le réseau.
2. **Injection de nœuds malveillants :** l'attaquant peut ajouter des nœuds malveillants dans le réseau pour injecter des données falsifiées.
3. **Le mauvais fonctionnement d'un nœud :** ceci peut générer des données inadéquates qui peuvent toucher l'intégrité du réseau.
4. **Trou noir (sinkhole) :** le nœud malveillant se place à côté de la station de base et supprime les messages qu'elle doit transmettre. Cette attaque devient fatal lorsqu'il n'y a qu'une seule station de base dans le réseau.
5. **Attaque par inondation avec le message HELLO :** de nombreux protocoles de routage utilisent des paquets "HELLO" pour découvrir leurs voisins et connaître la topologie du réseau. La plus simple attaque consiste à envoyer un flot de tels messages pour saturer le réseau et empêcher d'autres messages d'être échangés.
6. **Brouillage radio :** un attaquant va envoyer des ondes sur la même fréquence que le réseau [67], et les nœuds ne pourront plus communiquer car le médium est saturé par le brouillage radio [69].
7. **La privation de mise en veille :** cette attaque a pour but d'obliger un capteur de ne pas se mettre en veille par différents moyens [68], pour qu'il consomme brièvement sa batterie, jusqu'à se retrouver hors service [69].
8. **Insertions de boucles infinies :** un attaquant va modifier le routage du réseau avec un ou plusieurs nœuds malicieux, dans le but d'envoyer des messages qui vont être routés en boucles infinies et vont donc consommer l'énergie du réseau [69].

3.5 Les solutions de sécurité existantes pour les RCSF

Afin de remedier aux attaques cités,la cryptographie est une solution qui permet de chiffrer les données en utilisant les clés, et parmi les mécanismes fournis par la gestion de clés de cette techenique, nous citons [3, 6, 7, 67, 83, 68, 69] :

1. **Une seule clé :** partagée par tous les nœuds de réseau. Ce qui permet une utilisation efficace de la mémoire car le capteur a besoin de sauvegarder uniquement une seule clé, mais cette solution ne tolère pas la corruption de nœuds ; un nœud compromis par un attaquant signifie la déstruction totale du mécanisme de sécurité.

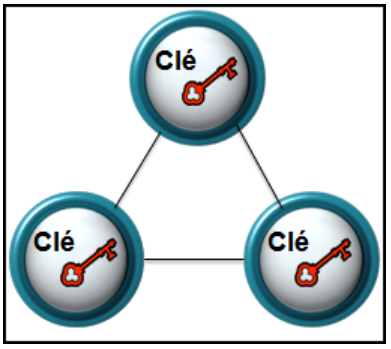

FIGURE 3.1 – Une seule clé partagée par tous les nœuds

2. **Des clés partagées par paire de nœuds :** chaque nœud est pré-chargé avec n-1 clés secrètes, chacune de ces clés est connue seulement par ce nœud et un des n-1 autres nœuds. Cette solution résiste aux compromissions de nœuds et assure l'authentification entre les paires. Cependant elle est convenable pour des petits réseaux, car elle exige extrêmement de mémoire, ainsi elle est non scalable, l'ajout ou la suppression des nœuds pose problème.

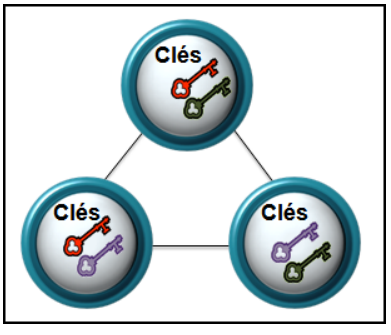

FIGURE 3.2 – Des clés partagées par paire de nœuds

3. **Clé publique :** Ce mécanisme de gestion de clés est difficilement concevable dans les RCSF à cause de sa lenteur en calculs, cependant, des recherches récentes ont montré qu'il est possible d'appliquer la solution à clé publique aux RCSF en choisissant les bons algorithmes et

les paramètres appropriés.

4. **Basée sur la station de base :** Ce mécanisme permet l'établissement de clé entre deux nœuds à l'aide de la station de base qui est considérée comme tierce partie de confiance avec laquelle chaque nœud partage une clé secrète. Pour que deux nœuds puissent communiquer entre eux d'une manière sécurisée, la station de base transmet une clé à chacun de ces nœuds en utilisant la clé secrète partagée avec eux.
   Cette solution présente les avantages suivants :
   – Une connectivité totale, où chaque nœud peut partager une clé avec n'importe quel autre nœud du réseau.
   – Une résistance contre la compromission d'un nœud.
   Notament, cette solution n'est pas appropriée aux RCSF à cause du nombre de messages requis, entre la station de base et les nœuds capteurs, afin d'installer des clés entre deux nœuds (temps de communication est trés grand).

3.6 Conclusion

Les propriétés des réseaux de capteur, certes ils permettent une grande facilité de production et de déploiement, mais rendent le système global de communication assez "fragile" à un certain nombre de défaillances. Afin d'assurer un déploiement à large échelle de cette technologie, il est nécessaire de pallier ces problèmes de sécurité aux différents niveaux d'une architecture RCSF. Et pour cela dans le chapitre suivant nous ferons une étude sur la cryptosystème de chiffrement asymétrique (à clé publique) ainsi une comparaison globale sur les différents critères afin de déterminer un algorithme adéquat pour les RCSF.

CHAPITRE 4

TAXONOMIE DES SYSTÈMES DE CHIFFREMENT À CLÉS PUBLIQUES

La conception et l'étude des cryptosystèmes de chiffrement à clés publiques spécifiques aux réseaux de capteurs sans fil a attiré une grande part d'intention des chercheurs de ce domaine. Ceci, car d'une part ces cryptosystèmes peuvent varier suivant leurs applications et l'architecture du réseau déployé, et d'autre part, ils doivent surmonter certains défis inhérents qui distinguent ce type de réseau, tels que un cryptosystème est considéré adaptatif si certains paramètres peuvent être contrôlés afin de s'adapter aux conditions et contraintes existantes du réseau et à l'énergie disponible, ainsi qu'aux ressources limités de stockage. En général, les cryptosystèmes de chiffrement asymétrique peuvent être divisé selon la structure du leur problèmes en plusieurs classes telles que nous trouvons des cryptosystèmes basés sur : la factorisation, le logarithme discret, le sac à dos, la théorie des codes, et le plus court vecteur non nul, etc. En outre, ces classes peuvent aussi contenir des sous-classes selon leurs principes de fonctionnement. Au cours de cette partie, nous proposons une taxonomie des cryptosystèmes selon la structure de leurs problèmes et leurs modes de fonctionnement, ainsi nous étudierons les différents cryptosystèmes afin de pouvoir les comparer et enfin élire le plus approprié pour les RCSF.

4.1 Taxonomie des cryptosystèmes asymétriques

Les algorithmes cryptographiques asymétriques connus à l'heure actuelle sont extrêmement variés ce qui rend assez difficile la tâche de les structurer en classification unifiée. Différentes taxonomies partielles sont toujours possibles, conformément aux critères concrets de classification. Dans la figure(4.1), nous avons classifié les déférents cryptosystèmes de chiffrement asymétrique selon le problème dont lequel leur principe est basé :

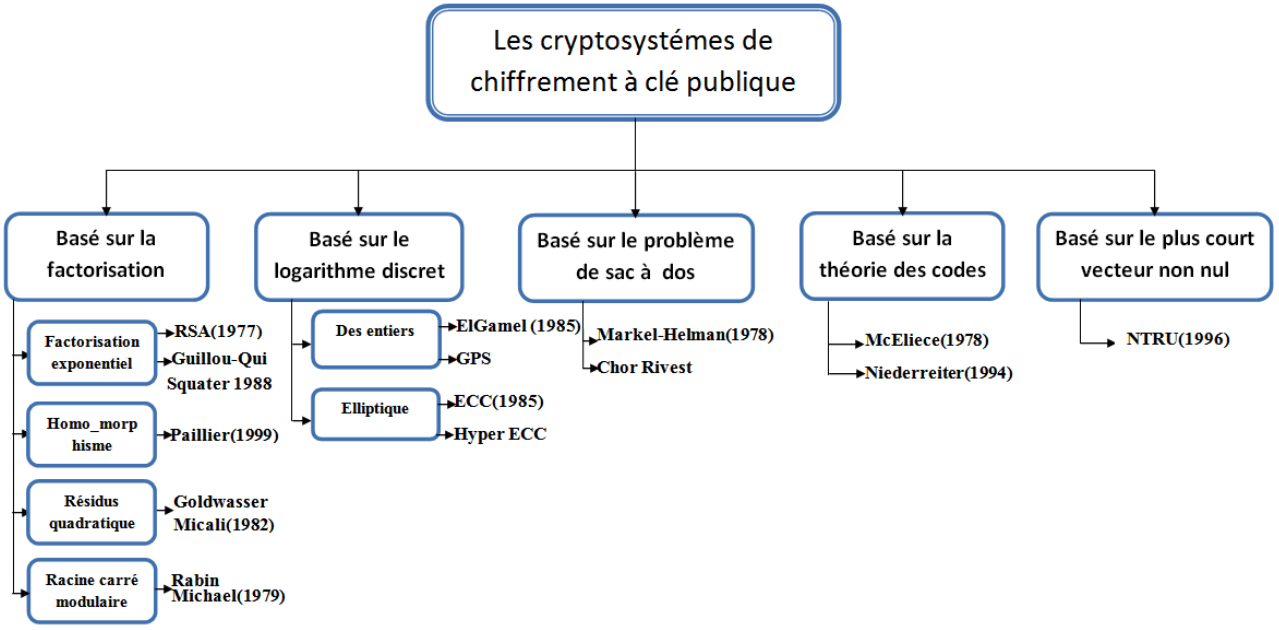

FIGURE 4.1 – La taxonomie des cryptosystèmes de chiffrement asymétrique

4.1.1 Les cryptosystèmes basés sur la factorisation

Le problème de la factorisation est récurrent en mathématiques et algorithmiques, c'est un problème qui peut s'exprimer de manière relativement simple, mais qui n'a pas, jusqu'à présent, de solution vraiment efficace [58, 95]. Toutefois, son application la plus visible est très certainement liée au nombreux cryptosystèmes différents, le but n'est pas de les énumérer tous mais plutôt de proposer ceux qui sont les plus efficaces et plus utilisés, tels que RSA, Guilou-Quisquater, Paillier-Pascal, Goldwaser-Micali, Rabin-Micheal.

4.1.1.1 RSA

Le cryptosystème RSA spécifie une manière de chiffrer et de signer des messages électroniques [58], ce qui permet de communiquer de façon authentifiée. Il est de nos jours largement utilisé pour sécuriser des communications, notamment dans le cadre du commerce électronique, créé en 1977 par Rivest, Shamir et Adleman [11, 24, 25, 78].

- **Principe**
  Il est basé sur la création d'une clé publique qui est diffusée, utilisée pour chiffrer le message et d'une clé privée gardée secrète utilisée pour déchiffrer le message [22, 26, 28, 30, 31] :
  1. **Génération de clés**
     – Choisir deux nombres premiers p et q, aléatoirement, p≠q.

- Calculer n = p∗q et φ(n)=(p-1)∗(q-1).
- e un entier choisi aléatoirement tel que 1 < e < φ(n) et pgcd(e,φ(n))=1
  (e et φ(n) sont premier entre eux.
- Calculer d tel que e∗d mod φ(n)=1.
- d est calculé en utilisant l'algorithme d'Euclide étendu, et ceci revient a résoudre une équation diophantienne.
- (e,n) est la clé publique et (d,n) est la clé privée.

2. **Chiffrement**
   - L'entier m est le message à chiffrer tel que 1<m<n , et c est le chiffré, calculé comme suit : c= m^e mod n.
   - Ce cryptogramme sera envoyé au récepteur concerné.
3. **Déchiffrement**
   Le récepteur à son niveau déchiffre ce cryptogramme à l'aide de sa clé privée *d*, comme suit : m = c^d mod n.

- **Exemple**
  Soit p=23 , q=19 , e=13
  Donc n=p∗q=23∗19=437 et φ(n)=22*18=396
  e * d mod φ (n)=1 $\rightarrow$ e * d - k * φ(n)=1 $\rightarrow$ d=61 (résolution d'équation diophantienne.
  Soit le message à chiffrer m= 309
  Le chiffré c=m^e mod n=309^{13} mod 437 =245.
  Pour déchiffrer ce cryptogramme m=245^{61} mod 437 =309.

- **Preuve de RSA**
  La formule de déchiffrement de c est :
  a=c^d mod n =m ?, remplaçant c par son équivalence c=m^e mod n
  a= ((m^e mod n)^d) mod n
  a= m^{e*d} mod n =m, puisque e est un inverse de d modulo n.

- **Complexité**
  Le chiffrement et le déchiffrement de RSA nécessitent une exponentiation modulo n. La taille de *(d,e)* est celle de φ(n), soit celle de n puisque φ(n)=(p-1)∗(q-1). Cela coûtent O (Log n)^3 multiplications modulo n.
  Cette complexité n'est pas négligeable, notamment pour des environnements à puissance de calcul réduite. Cela rend l'algorithme lent en termes de calcul et en espace mémoire.

4.1.1.2 Guillou Quisquater

Ce cryptosystème est un système de chiffrement à clés publiques inventé par Louis Guillou et Jean-Jacques Quisquater en 1988, il est basé sur le problème de factorisation. Il utilise une exponentiation de RSA, et il est utilisé dans nos jours dans les cartes à puces [43, 55, 56, 82].

- **Principe**
  - L'entité A veut s'authentifier auprès de L'entité B.
  - n = p∗q, de la même manière que dans RSA.
  - φ(n)=(p-1)∗(q-1).
  - v un nombre qui sert comme une clé publique tel que pgcd(v, φ(n))=1.
  - s un nombre qui sert comme clé privée tel que : s ∗ v mod φ(n) = 1.
  - Ja un certificat public, Sa un certificat privé tel que Sa= Ja^{-s} mod n.
  - A choisit un nombre aléatoire m.
  - A calcule x=m^v mod n.
  - A envoie x et Ja à B.
  - B choisit un nombre aléatoire e (défi) tel que 1< e < v.
  - B envoie e à A.
  - A calcule y=m∗ Sa^e mod n , et l'envoie à B.
  - B calcule Ja^e ∗ y^v mod n , et vérifie que le résultat est égal à x et $\neq$ 0.
- **Exemple**
  soit p = 7, q = 11.
  - n = p∗q = 7∗11= 77.
  - φ(n)= (p-1)∗(q -1) = 6∗10=60.
  - Nous générons aléatoirement l'exposant de chiffrement v=13,et il faut juste qu'il soit inversible au mod φ =60 pour qu'on puisse calculer s :
    v∗s mod φ(n)=1 $\longrightarrow$ s = 37.
  - Soit Ja=4 $\longrightarrow$ Sa=9.
  - L'entité A choisit aléatoirement un nombre m et calcul son chiffré x, x = m^v mod n :
    m =26, x=26^{13} mod 77= 75.
  - L'entité B choisit un entier 1 <e <13(le défi entre les deux entités), et le transmet à A : soit e = 8.
  - L'entité A calcule Y = m∗Sa^e mod n, et le transmet à B :
    Y=26∗9^8 =34.
  - Et finalement B vérifie si Ja^e ∗ y^v= 4^8 ∗ 35^{13} = 75 = x. Légalité est vérifiée.
- **Preuves de Guillou Quisquater**
  L'emetteur calcule x=m^v mod n , y=m∗Sa^e mod n
  Et le recepteur à son niveau vérifie si : Ja^e ∗ y^v mod n=x ? a=Ja^e ∗ y^v mod n
  a=Ja^e ∗ (m∗ Sa^e mod n) ^{v} mod n

a=Ja^{e} $*$ m^{v} $*$ Sa^{ev} mod n
a=Ja^{e} $*$ m^{v} $*$ Ja^{-sev} mod n
a=(Ja$*$ Ja^{-} mod n) ^{sev} $*$ m^{v} mod n
a=(1) ^{sev} $*$ m^{v} mod n , car Ja et Ja^{-} sont des inverses
a=m^{v} mod n=x.

- **Compléxité**
  Le calcul de x= m^{v} mod n, tel que v< n, on nécessite une complexité de O(log n)^{3}, car on dispose de modulo exponentielle, et pour calculer la complexité de l'opération (y=m $*$ Sa^{e}) mod n qui égale au produit des deux complexités (O log n)^{2} $*$ O(log n)^{3}.
  Pour le déchiffrement on a besoin de calculer la complexité de produit de Ja^{e} $*$ y^{v} ce qui nous donne la complexité de O(log n)^{2}.
  Pour les exposants v et s, leur taille dépendra de φ(n),et soit celle de n puisque φ(n)= (p- 1)$*$(q -1).
  Nous concluons que le chiffrement et le déchiffrement de Guillou Guisquater coûtent O(log n)^{3} multiplications modulo n.

4.1.1.3 Paillier

Le cryptosystème de Paillier conçu par Pascal Paillier en 1999, basé sur un homomorphisme additif [84, 88, 91].

- **Principe** [104, 107, 108]

  1. **Génération de clés**
     - Choisir aléatoirement deux nombres premiers p et q tels que p$\neq$q.
     - Calculer la clé publique n= p$*$q et la clé privée φ(n) = (p -1)$*$(q - 1).
     - Soit r, un entier aléatoire tel que : 0<r<n
     - Calculer r^{-} tel que r$*$ r^{-} mod n=1.
  2. **Chiffrement**
     - Soit m un message à chiffrer tel que : 0 <m< n
     - Le message chiffré est alors : c= (1+ n)^{m} * r^{n} mod n^{2} .
  3. **Déchiffrement** Pour retrouver le texte clair m :
     m=$\frac{(c*r^{-n} mod n^{2})-1}{n}$.

- **Exemple**
  soit p=7, q=11.
  n=p$*$q=7$*$11=77, φ(n)=(p-1)$*$(q-1)=6$*$10=60.

m le message à chiffrer (0<m<77), on prend m=12.
r un exposant de chiffrement (0<r<77), on prend r=15.
Ensuite on calcule l'inverse de r tel que pgcd(r,n)=1 $\longrightarrow$ r^{-}=36.
pour chiffrer le message m :
c=$((1+n)^m * r^n) \bmod n^2$=$((1+77)^{12} * 15^{77}) \bmod 77^2$=533.
pour déchiffrer le cryptogramme c :
On calcul m comme suit : m= $\frac{(c*r^{-n} \ mod \ n^2)-1}{n}$ = $\frac{533*36^{77} mod 77^2-1}{77}$= $\frac{924}{77}$=12.

- **Preuves de Paillier**
  - ▷ *La formule de déchiffrement est* :
    a= $\frac{(c*r^{-n} \ mod \ n^2)-1}{n}$=m ?
    Remplaçant c par sa formule qui représente le chiffré de m tel que c=$((1+n)^m * r^n) \bmod n^2$, on aura :

    a= $\frac{((1+n)^m * r^n) mod n^2 * r^{-n} \ mod \ n^2)-1}{n}$

    a= $\frac{((1+n)^m * ((r^n * r^{-n}) mod n) \ mod \ n^2)-1}{n}$

    On a $((r^n * r^{-n}) \bmod n)=1$, donc :

    a= $\frac{(((1+n)^m mod \ n^2)-1}{n}$, et cela en utilisant la formule
    α : $(1+n)^x = 1 + x*n \bmod n^2$ [104].

    a=1+n*m mod n^2 -1 =$\frac{n*m}{n} mod n^2$= m

    Donc $\forall$ m , si c=cryptage-Pailler(m) $\longrightarrow$ décryptage-Pailler(c)=m.

  - ▷ *Paillier est basé sur l'homomorphisme* :
    En d'autres termes, avec uniquement la clé publique et le chiffrement de m_1 et m_2, il est possible de calculer le chiffrement de m_1+m_2.

    Soit : c' le chiffré de m' et c" le chiffré de m", en utilisant la même clé publique(n).
    c' =$(1+n)^{m'} * r^n \bmod n^2$ et c"" =$(1+n)^{m"} * r^n \bmod n^2$
    c'+c"=$(1+n)^{m'} \bmod n^2$ + $(1+n)^{m"} \bmod n^2$
    c'+c"=$(1+n)^{m'} * r^n \bmod n^2$ + $(1+n)^{m"} * r^n \bmod n^2$
    c'+c"=$r^n * ((1+n)^{m'}+(1+n)^{m"})$
    Donc :
    c'+c"=$r^n * (1+n* m'+1+n* m" \bmod n^2$
    c'+c"=$r^n * (1+n* m'+1+n* m" \bmod n^2$

c'+c"=r^n $*$ (1+1+n$*$ (m'+m")mod n^2
D'aprés α
c'+c"=r^n $*$(1+ n$*$ (1+n)$^{(m'+m")}$)mod n^2
Nous nottons c le chiffré de m :
c'+c"=r^n $*$(1+ c)mod n^2 $\longrightarrow$ c=c'+c" -1.

- **Compléxité**
  La complexité de chiffrement dépend de la complexité d'exponentiation modulaire de la formule suivante :
  $((1+n)^m * r^n) \mod n^2$, qui égale au $O(\log n)^3$.
  Par contre, la complexité de déchiffrement est égale à la complexité de la formule suivante : $\frac{(c*r^{-n} mod n^2)-1}{n}$ qui est égale à $O(\log n)^3$.

4.1.1.4 Goldwasser Micali

Le cryptosystème de Goldwasser-Micali (GM), est développé par Shafi Goldwasser et Silvio Micali en 1982. C'est le premier cryptosystème à chiffrement probabiliste qui est prouvablement sûr, avec des hypothèses cryptographiques standards. Toutefois, il n'est pas efficace : les textes chiffrés peuvent être des centaines de fois plus longues que les textes d'origine. Afin de prouver la sécurité de ce cryptsystème, ses auteurs ont proposé la définition de sécurité sémantique qui est, de nos jours, largement utilisée [55, 61, 73, 81, 117].

- **Principe**
  Chaque entité crée une clé publique et une clé privée correspondante, comme suit :

  1. **Génération de clés**
     - Sélectionner deux grands nombres premiers p et q aléatoirement tel que p$\neq$q.
     - Calculer n = p$*$q.
     - Sélectionner z $\in$ Z_n tel que z est un résidu quadratique non modulo n et le symbole de Jacobi : (z/n) = 1.
     - La clé publique est (n,z), et la clé privée est la paire (p,q).
  2. **Chiffrement**
     Pour que B peut chiffrer un message m pour A, il doit faire ce qui suit :
     - obtenir une clé publique (n,z).
     - Représenter le message m comme une chaîne binaire m = $m_1 m_1 ... m_t$ de longueur t.
     - Pour i allant de 1 à t faire :
       a) Choisir un $r_i \in Z_n$ au hasard.
       b) Si m_i = 1 alors $c_i \longrightarrow z * r_i^2 \mod n$; sinon $c_i \longrightarrow r_i^2 \mod n$.
     - Envoyer le t-uplet c = (c_1, c_2,...,c_t) à A.

3. **Déchiffrement**
   Pour récupérer le clair m de c, A doit faire ce qui suit :
   – Pour i allant de 1 à t faire :
     a) Calculer le symbole de Legendre e_i= (c_i/p)
     b) Si e_i = 1 alors $m_i \longrightarrow 0$, sinon $m_i \longrightarrow 1$.
   – Le message déchiffré est m = $m_1 m_2 ... m_t$.

- **Exemple**
  Soient p = 7, q = 3 et donc n = 7∗3 = 21.
  Nous cherchons un z ∈ Zn, qui soit un résidu non quadratique modulo n et tel que (z/n) = 1
  Les résidus quadratique modulo 21 sont {1,4,7,9,15,16,18} (ce sont les seuls résultats possibles de la mise au carré des éléments de Zn modulo n).
  Prenons z = 11 (qui est bien un résidu non quadratique modulo 21) et calculons le symbole de Jacobi : $\frac{11}{21}$= $\frac{11}{7} * \frac{11}{3}$ = (11^1 mod 3)∗ (11^3 mod 7) = -1 ∗ 1 = -1.
  Et donc z = 11 ne convient pas.
  Essayons z = 5 (qui est aussi un résidu non quadratique modulo 21) :
  (5/21) = (5/3)*(5/7) = (5^7 mod 3) *(5^3 mod 7) = -1∗ -1 = 1.
  Donc z = 5 convient.
  Si une entité A veut chiffrer le message m = 10110 pour entité A, elle choisit au hasard :
  r_1 = 4, r_2 = 8, r_3 = 13, r_4 = 5 et r_5 = 4. Puis elle calcule :
  c_1 = 5 *4^2 mod 21 = 80 = 17
  c_2 = 8^2 mod 21= 1
  c_3 = 5 * 13^2 mod 21 = 5
  c_4 = 5 *5^2 mod 21= 20
  c_5 = 4^2 mod 21 = 16
  Et donc c = (17, 1, 5, 20, 16).

  Pour déchiffrer, l'entité B évalue les symboles de Legendre suivants :
  (c_1/p) =(17/7) = 17^3= 4913 = -1 mod 7 $\neq$ 1 $\longrightarrow$ m_1 = 1.
  (c_2/p) = (1/7) = 1^3 = 1 mod 7 $\longrightarrow$ m_{22} = 0.
  (c_3/p) = (5/7) = 5^3 = 125 = -1 mod 7 $\neq$ 1 $\longrightarrow$ m_3 = 1
  (c_4/p) = (20/7) = 20^3 = 8000 = -1 mod 7 $\neq$1 $\longrightarrow$ m_4 = 1
  (c_5/p) = (16/7) = 16^3 = 4096 = 1 mod 7 $\longrightarrow$ m_5 = 0
  Et finalment m =$m_1 m_2 m_3 m_4 m_5$= 10110.

- **Complexité**
  Pour calculer la complexité de cet algorithme, on calcule la complexité de chiffrement et de déchiffrement, tel que :
  – La complexité de chiffrement est celle des opérations de la boucle de l'étape (b) de l'algorithme de chiffrement qui égale à O(log n)^3.
  – Et de même pour le déchiffrement, tel que sa complexité dépend des opérations de la boucle

(a) de l'algorithme de déchiffrement et qui égale à la complexité de l'opération (e_i =C_i / p) et d'après le tableau 27, on constate qu'elle est égale à O(log n)^2.

Nous concluons que le chiffrement et le déchiffrement de Goldwasser-Micali coutent O(log n)^3 multiplications modulo n.

4.1.1.5 Rabin Micheal

Cet algorithme fut publié en 1979 par Michael Rabin, basé sur le problème de la racine carrée modulaire, qui est équivalent à la factorisation. Cependant, cette équivalence est influencée par des critères qui peuvent sembler incompatibles avec les aspects pratiques. Comme on risque d'avoir plusieurs racines valides pour un seul chiffre [42, 45, 49].

- **Principe** [85] :

  1. *Génération de clé avec Rabin Micheal :*
     Soit n=p$*$ q,tels que p et q sont deux nombres premiers distincts et sont congrus à 3 modulo 4(p modulo 4=3, q modulo 4=3).

     Dans le cas de chiffrement des lettres, on le décompose en blocs, et le symbole * débutera chaque bloc de texte, et cela permet de déterminer le message clair approprié parmi les quatre trouvés.
  2. *Chiffrement avec Rabin Micheal :*
     Soit m le message à chiffré (0 <m$\leq$ n-1), son déchiffrement s'obtient comme suit : c=m^{2}mod n.
  3. *Déchiffrement avec Rabin Micheal :*
     Le message clair m s'obtient à partir du cryptogramme c, par la formule :
     m=c$^{\frac{1}{2}}$ mod n , qui sera calculé en utilisant l'algorithme des restes chinois tel que l'entité b trouve deux racines carré comme suit :
     p_1=c$^{\frac{1}{2}}$ mod p =4$^{\frac{p+1}{4}}$ mod p
     q_1=c$^{\frac{1}{2}}$ mod q =4$^{\frac{q+1}{4}}$ mod q
     Pareillement, elle calcule : p^-,q^- qui sont des inverses de p et q mod n dans cet ordre, en utilisant l'algorithme d'Eclide étendu.
     Finalement, elle calcule les quatre messages correspondants au cryptogramme c :
     m_i = $\pm$ p $*p^-*q_1 \pm$ q $*q^-*p_1$ mod n

- **Exemple**
  - Soient p = 7 et q = 11
    Ces deux valeurs sont des clés privées et ne doivent pas être rendues publiques. On calcule ensuite n = 7$*$11= 77 qui constitue la clé publique.

– La fonction de chiffrement est : c=m^2 mod n
L'entité A désire envoyer le message m=2 à l'entité B.
Le chiffré c =m^2 mod n= 2^2 mod 77= 4, donc A transmis 4 à B.

– La formule de déchiffrement est : m=$c^{\frac{1}{2}}$ mod n
P' = $4^{\frac{1}{2}}$ mod 77 .....(a)
Dans ceci la résolution de l'équation (a) mène à résoudre le système d'équations en utilisant l'algorithme des restes chinois.

p_1=$4^{\frac{1}{2}}$ mod 7 =$4^{\frac{7+1}{4}}$ mod 7 =4^2 mod 7=2
q_1 = $4^{\frac{1}{2}}$ mod 11 = $4^{\frac{11+1}{4}}$ mod 11 =4^3 mod 11=9
En outre il calcule : p^-,q^- qui sont des inverses de p,q dans cet ordre :
P^- = 7^{-1} mod 11 = 8
q^- = 11^{-1} mod 7 = 2
Les quatre messages, correspondants au cryptogramme c sont obtenus comme suit :
m_1=(7∗8∗9 + 11∗2∗2)mod 77=9
m_2=(-7∗8∗9 - 11∗2∗2)mod 77=68
m_3=(7∗8∗9 - 11∗2∗2)mod 77=75
m_4=(-7∗8∗9 + 11∗2∗2)mod 77=2
Et finalement, L'entité B choisit le bon message, parmi ces derniers.

- **Complexité**
La complexité d'algorithme de Rabin Michael est O Log(n)^2. Le seul calcul coûteux de cet algorithme est le calcul de multiplications modulo n et de la puissance m^2 mod n (à effectuer au plus k fois).

4.1.2 Les cryptosystèmes basé sur le logarithme discret

le problème du logarithme discret peut se formuler dans n'importe quel groupe. Soit G un groupe de cardinal n,et Soit a $\in$ G.On appelle logarithme discret le problème de trouver un unique élément x tel que : g^x = a. Nous proposons ici d'étudier les deux cryptosystèmes basés sur le logarithme discret : Elgamal et les courbes elliptiques[41].

4.1.2.1 El-Gamal

Cet algorithme de chiffrement à clés publiques a été inventé par Tahar El Gamel en 1985, en effet il est lié au problème de logarithme discret qui consiste à retrouver un entier x tel que :

A= B^x mod C [29, 33, 35, 36, 87] :

- **Principe**

  1. **Génération de clés**
     - Choisir deux entiers p et g tel que p premier, p et g sont premiers entre eux.
     - Choisir la clé secrète s tel que s<p.
     - Calculer la clé publique b=g^s mod p.
  2. **Chiffrement**
     - Soit un message m<p, on choisit un entier k qui n'est connu que par l'émetteur.
     - Le chiffré c= (c_1,c_2) tel que c1=g^k mod p, et c2=m*b^k mod p.
  3. **Déchiffrement**
     Calcul de r_1=$c_1{}^s$ mod p. et m=c_2/r_1.

- **Exemple**
  Soit p=13 ; g=3 ;s=8 ;k=5 ;m=4 b=g^s mod p=3^8 mod 13=6561 mod 13=9.
  Chiffrement :
  c_1=g^k mod p= 3^5 mod 13=9.
  c_2=m*b^k mod p=4*9^5 mod 13=12.
  Déchiffrement :
  r_1=$c_1{}^s$ mod p=9^8 mod 13=3.
  m=c_2/r_1=12/3=4.

- **Preuves d'El-Gamal**
  Dans le cryptosystème d'Elgamal :
  c2=m*b^k mod p et r_1=$c_1{}^s$ mod p et b=g^s mod p
  a=$\frac{c_2}{r_1}$ =m ?

  a=$\frac{c_2}{r_1}$ =$\frac{m*b^k modp}{c_1^s modp}$

  a=$\frac{m*g^{sk} modp}{g^{ks} modp}$=m

- **Complexité**
  La complexité de chiffrement est celle d'exponentiel modulo P, tel que le chiffrement et le déchiffrement coûtent O Log(n^3) multiplications modulo p.

4.1.2.2 Le cryptosystème des courbes elliptiques

La théorie des courbes elliptiques est très ancienne, mais son utilisation en cryptographie date de 1985 qui à été proposée par Neal Koblitz [74] et Victor S. Miller [75].
L'idée était de transformer les cryptosystèmes utilisant des groupes finis du type F∗p en cryptosystèmes utilisant l'arithmétique des courbes elliptiques [72].
L'opération de base est la multiplication de points d'une courbe avec de grands nombres entiers : k*P, qui revient en effet à une addition du point P, k fois [122]. La cryptographie par courbe elliptique pose l'épineux problème de la résolution du logarithme discret [120]. La sécurité de cette méthode cryptographique réside dans le choix de la courbe elliptique et de la difficulté que celle-ci pose à résoudre le problème du logarithme discret [109, 121].

Une courbe elliptique définie sur K peut être caractérisée par son équation Weierstrass.
E : $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_3x + a_6$;

La figure 4.2 illustre l'application de la méthode de Diffie-Hellman aux courbes elliptiques [120].

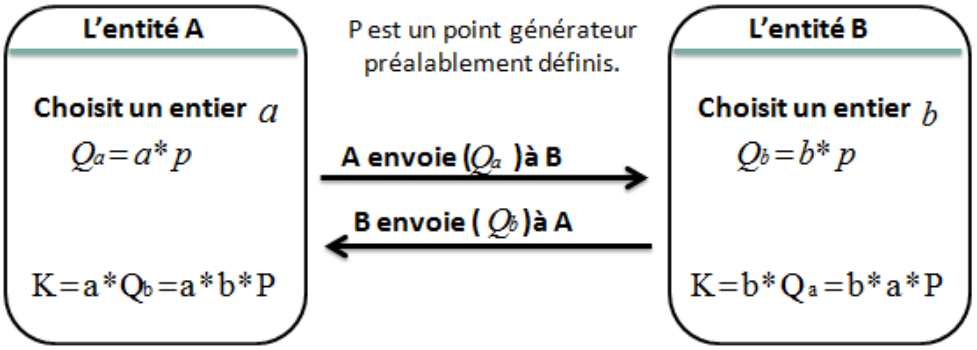

FIGURE 4.2 – Application de la méthode de Diffie-Hellman aux courbes elliptiques

- **Principe**
  La méthode classique de Diffie-Hellman, est basée sur un groupe multiplicatif modulo p. Celle utilisée avec les courbes elliptiques appelées ECDH (cf. figure 4.2) est basé sur un groupe additif des points d'une courbe elliptique.
  Soit la courbe elliptique d'équation : [50]
  E : $y^2 + y = x^3 - x$.
  Utilisation de la transformation :
  x=x' et y=y'-376
  D'ou E' : $y^2 = x^3 - x + 188$

Définit sur un corp fini IF_q, tels que p=751.

Le chiffrement

Soit m=s est le message à chiffrer, par la suite des points de la courbe E.

Avant le chiffrement on effectue le codage suivant : les chiffres de 0 à 9 sont codés de 0 à 9, ainsi les lettres de a à z sont également codés de 10 à 35. Le "s" est codé par 28

Donc x=m$*$k+j avec k=20 ou 30 ou 50 et 1$\leq$ j$\leq$ k

On pose k=20.

pour j=1 : x=28*20+1=561

On remplace dans E' on trouve :

y^2=(176558108)mod p=261

Cherchons si 261 est un carré d'un nombre modulo p, et cela en utilisant le symbole de jacobi.

$(\frac{261}{751})$=(-1)^{230*375} $(\frac{751}{261})$=$(\frac{751}{261})$

$(\frac{751}{261})$=$(\frac{229}{261})$=(-1)^{114*130}=$(\frac{261}{229})$=$(\frac{261}{229})$

$(\frac{261}{229})$=$(\frac{32}{229})$=$(\frac{2^5}{229})$=$(\frac{2}{229})^5$

$(\frac{2}{229})$=(-1)$^{\frac{229^2}{8}}$ =-1

D'ou :

$(\frac{2}{229})^5$=-1.

Donc 261 n'est pas un carré modulo 751.

On incrémente j, j=2, on trouve x=562.

On remplace dans E' on trouve :

y^2=(177503954)mod p =598

Cherchons si 598 est un carré d'un nombre modulo 751.

$(\frac{598}{751})$=$(\frac{2}{751})(\frac{299}{751})$

$(\frac{2}{751})$=(-1)$^{\frac{751^2}{8}}$=1

$(\frac{299}{751})$=-$(\frac{751}{299})$

-$(\frac{751}{299})$=-$(\frac{153}{299})$=-$(\frac{299}{153})$

-$(\frac{299}{153})$=-$(\frac{146}{153})$=-$(\frac{2}{153})(\frac{73}{153})$

$(\frac{2}{153})$=(-1)$^{\frac{153^2}{8}}$=1

-($\frac{73}{153}$)=-(-1)^{36*76} =-($\frac{153}{73}$)

-($\frac{153}{73}$)=-($\frac{7}{73}$)=-($\frac{73}{7}$)

-($\frac{73}{7}$)=-($\frac{3}{7}$)=-(-1)^3($\frac{7}{3}$)=($\frac{7}{3}$)

($\frac{7}{3}$)=($\frac{1}{3}$)=(-1)^{0*1}=(-1)^0=1

Donc 598 est un caré modulo 751.
Cherchons la racine carrée modulo 751 de 598
En utilisant la loi de réciprocité quadratique qui est le symbole de Jacobi, on détermine si un entier a est un carré ou non modulo un nombre premier p, (c'est à dire il existe un x tel que x^2mod p=a).
On suppose n un résidu non quadratique de p, soit n=3.
On écrit p-1 sous la forme $2^\alpha * s$, oú s est impair.
On calcule b=n^s mod p
On calcule y=$a^{(s+1)/2}$ mod p=$598^{(375+1)/2}$ mod 751=201
Cette valeur de y est celle dans E', en utilisant la transformation on trouve y=222
Donc le point(562,222) représente la lettre s.

Déchiffrement
On a x=m∗k+j $\longrightarrow$ m=$\frac{x-j}{k}$
x=262 ;j=2 ; k=20 $\longrightarrow$ m=$\frac{262-2}{20}$=28, et cela coresspand a la lettre s.

- **Complexité**
  La méthode des courbes elliptiques en cryptographie requiert un temps :
  $2^{((logn)\frac{1}{2}(log(logn))\frac{1}{2})}$ [18].

4.1.3 Les cryptosystèmes basés sur le sac à dos

Nous proposons d'étudier le cryptosystème à clés publiques de Merkle-Hellman qui est basé sur le problème du sac à dos (Knapsack problem) qui un problème NP complet.

4.1.3.1 Merkel Hellman

En 1978, Ralph Merkle et Martin Hellman proposèrent ce cryptosystème à clés publiques qui utilise des instances particulières du problème de sac à dos[20] dont ils l'ont transformé en un pro-

blème résoluble en un temps polynomial en utilisant la notion de suite super croissante [97].
Ce cryptosystème est un à sens unique ; la clé publique est utilisée uniquement pour le chiffrement, et la clé privée uniquement pour le déchiffrement. Il ne peut donc pas être utilisé comme un protocole d'authentification [18, 38, 39, 40] car il n'admet pas de signature numérique.

- **Principe de Merkle-Hellman** Le problème du sac à dos consiste à empiler des objets dans un sac, de manière à atteindre (si possible) un poids total fixé. Plus formellement, étant donnés des poids entiers P_1,...,P_n et un poids de total T, il s'agit de trouver b_1,... $b_n \epsilon\{0,1\}$, tels que : T = $b_1 * P_1$+$b_2 * P_2$ + ... + $b_n * P_n$.
  Il ne semble pas exister d'algorithme rapide permettant de résoudre ce problème [107].
  Cependant si la suite des poids P_k est super croissante alors le problème du sac à dos ne présente aucune difficulté, car il existe une méthode de résolution simple en utilisant algorithme de glouton.

  1. **Génération de clés**
     L'idée de base du système consiste à construire une suite non super croissante à partir d'une suite super croissante.
     Une entité A choisit :
     - Une suite super croissante S= (a_1, a_2,...,a_n) qui représente la clé privé.
     - Un nombre n supérieur à a_n
     - Un entier e, 1 < e < n, et premier avec n, c'est un exposant de chiffrement publique.
     - Pour chaque élément a_i de S, A calcule : $b_i = a_i * e \bmod n$, et trouve S'=(b_1, b_2, ... , b_3) qui n'est plus super croissante, qui sert comme clé publique utilisé dans le chiffrement .
     - d=inverse(e) modulo n (d trouvé par l'Euclide étendu.
  2. **Chiffrement**
     Pour chiffrer un message m, l'entité B le représente en code binaire et le décompose en blocs de même longueur, de tel façon que cette longueur sera inférieur à la taille de la suite afin d'éliminer le cryptanalyse par attaque exhaustive.
     Pour chaque bloc m_1 m_2...m_n, elle calcule $c_i = m_i * b_i$ oú $b_i \in$ S'. Finalement, l'émetteur regroupe le cryptogramme c=c_1 c_2...c_n.
  3. **Déchiffrement**
     Pour déchiffrer le cryptogramme c reçu, l'entité A calcule x_i= $c_i * d$ modulo n et détermine x_1, x_2, ... , x_n tels que $c_i * a_i = x_i$ (Il s'agit d'un problème simple de sac à dos, la suite (a_1, a_2, ... , a_n) étant super croissante.)
     Finalement le message m=m_1 m_2...m_n.

- **Exemple**

– L'entité A choisit une suite super croissante S= (2, 5, 11, 22, 42, 84) de 6 éléments, n=107 et e=29.
– Calcule de la clé privée d =inverse(e mod n) = inverse (29 mod 107) $\rightarrow$ d$*$e-n$*$a=1 , 29$*$d-107$*$a=1 , d= ?.
  Après l'application de l'algorithme d'Euclide étendue en trouvant d=48
– Calcule de la suite S' qui sert comme une clé publique :
  S' : (bi = ai $*$ e mod n), ce qui donne S'= (58,38,105,103,41,82).

Chiffrement :

– Soit m="sac" le message que l'entité B veut crypter afin de le transmettre à l'entité A.
– L'entité B utilise le tableau de codage (cf. tableau 4.1) afin de représenter le message sous format binaire. Nous avons supposé la taille de chaque bloc égal à cinq, et cela en rendant ce cryptosystème multi alphabétique pour éviter la cryptanalyse de texte chiffré.

a=00001	b=00010	c=00011	d=00100	e=00101	f=00110	g=00111
h=01000	i=01001	j=01010	k=01011	l=01100	m=01101	n=01110
o=01111	p=10000	q=10001	r=10010	s=10011	t=10100	u=10101
v=10110	w=10111	x=11000	y=11001	z=11011		

TABLE 4.1 – Tableau de codage binaire

– Le message à coder est alors "10011 00001 00011", et comme la clé publique comporte six éléments, donc B doit regrouper les bits de message par des blocs de 6 bits, en ajoutant des '0' à droite, $\rightarrow$ m= "100110 000100 011000".

– Maintenant B chiffre chacun des blocs : bloc =bit$*$ b_i , tel que b_i est un terme de suite s' :

  m_1=100110 $\Rightarrow$ c_1= 58+103 +41= 202.
  m_2=000100 $\Rightarrow$ c_2= 103.
  m_3=011000 $\Rightarrow$ c_3= 38+105=143.
  l'entité B transmet le message : "202, 103, 143" à l'entité A

Déchiffrement :

L'entité A déchiffre élément par élément, en utilisant l'exposant de chiffrement d ; donc elle calcule x_i= c_i $*$d mod n et détermine la solution du problème du sac à dos correspondant dans la suite super croissante S, comme suit :

x_1=202 $*$ 48 mod 107 = 66=42+22+2 $\Rightarrow$ m_1= 100110.
x_2=103 $*$ 48 mod 107 = 22 $\Rightarrow$ m_2=000100.
x_3=143 $*$ 48 mod 107 = 16=16+5 $\Rightarrow$ m_3=011000.

Enfin, l'entité A retrouve le message m= m_1 m_2 m_3="100110 000100 011000" et après la mise en ordre selon la taille des blocs supposés initialement $\Rightarrow$ m="100100 000000 010000"
Pour retrouver le texte, l'entité A n'a plus qu'à consulter le tableau 4.1 pour reconstruire le

message clair, qui correspond bien à "sac".

- **Complexité**

  L'algorithme de Merkel Hellman est exponentiel, de complexité O(e^n), cet algorithme relève de classe NP complet, mais il est transformé en problème facile et cela grâce la suite super croissance qui rend ce cryptosystème plus rapide, et ainsi minimise le temps de calcul et nombre d'opérations effectué.
  Cependant, sans cette transformation, le chiffrement et le déchiffrement nécessitent un calcul coûteux de exponentiation modulo n et ce calcul est fait plusieurs fois a des blocs de longueur n ce qui engendre cette complexité.

4.1.4 Les cryptosystèmes basés sur la théorie des codes

La théorie des codes, traite des codes et de leurs propriétés et leurs aptitudes à servir sur différents canaux de communication. Il existe des cryptosystèmes utilisant dans leur chiffrement les codes correcteurs, nous prenons l'un de ces cryptosystèmes tel que McEliece.

4.1.4.1 McEliece

Ce cryptosystème asymétrique, inventé en 1978 par Robert McEliece, repose sur le principe de la théorie des codes. Il n'est pas rencontré de véritable soutien dans la communauté cryptographique à cause de la taille de sa clé publique. Il possède deux propriétés importantes ; La sécurité augmente plus avec la taille des clés, ainsi sa rapidité de chiffrement [52, 110, 111, 125].

- **Principe**
  Son principe est basé sur l'ajout d'un code correcteur d'erreur, pour ce faire, nous prenons un mot, le transformons en un mot de code qui contient à la fin une information (appelée redondance), et à la sortie grâce à la redondance, nous éliminerons les erreurs et retrouvons le mot de code, puis avec la transformation de ce mot nous obtiendrons le mot original.
  L'idée derrière tous ça est d'ajouter autant d'erreurs possibles au message après transformation en code dans le but de masquer le code mais on laisse la possibilité de corriger l'erreur, et si cette méthode de correction est gardée secrètement, alors seul le destinataire sera en mesure de retrouver le message original.
  En outre, il utilise des codes de Goppa qui sont faciles dans le décodage de l'information [52].

  1. **Génération de clés**
     - Sélectionner aléatoirement C=(n,k)code linéaire capable de corriger t erreurs. Ce code doit posséder un algorithme de décodage efficace.

- L'entité A génère une matrice génératrice G (k*n) pour le code C.
- Sélectionner aléatoirement une matrice non singulière S binaire (k*k).
- Sélectionner aléatoirement une matrice de permutation P de dimension n*n.
- Calculer la matrice G'=S*G*P celle ci est une matrice (k*n).
- La clé publique est (G',t) ; la clé privée est (S,G,P).

2. **Chiffrement**

   Pour chiffrer le message binaire m de longueur k, il faut :
   - Calculer le vecteur c'=m$*$G'.
   - Générer un vecteur d'erreur e de poids t qui représente une suite binaire de longueur n.
   - Calculer le chiffré c=c'+e.

3. **Déchiffrement**
   - Calculer c'=c$*$P^{-1} tel que on dispose d'une matrice sympathique qui peut facilement éliminer le vecteur d'erreur dont elle contient les codes correcteurs, et utiliser l'algorithme de décodage pour extraire c' en un mot m'.
   - Calculer m=m'$^{S-1}$

- **Exemple**

  Soit C le code Goppa de paramètre [n,k,d]=[3,4,2] tel que : n= nombre de ligne de la matrice génératrice.

  k = nombre de colonnes de la matrice génératrice.

  d= c'est la distance de Haming qui représente ici le nombre maximum d'erreurs t.

  G la matrice génératrices : $G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

  Soit la matrice inversible $S = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \longrightarrow$ la matrice inversible $S^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

  La matrice de permutation $P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \longrightarrow p^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

  La matrice publique : $G' = SGP = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$

  Et Soit m=(101) le message à chiffrer.

  Donc le chiffré c' = M$*$G' =(1010)

On suppose que e=(0100) le vecteur d'erreur à ajouter.
c= c' + e =1100
L'émetteur envoie c à son récepteur.
Le récepteur reçoit c, et le déchiffre comme suit :
a=c$*$p^{-1}=(1001)
Extraire m' de a en utilisant les algorithmes de décodage des codes correcteurs.
m=m'$*$p^{-1}=(101)
On a :
a=c$*$p^{-1} = (mG'+e)$*$p^{-1} =(mSGP+e)$*$p^{-1} =mSG + e $*$ p^{-1} =m'G+ e'.

- **Complexité**
  Lorsque on calcule le produit de 3 matrices S*G*P=G' de taille (n*n),(n*k),(k*k) dans cet ordre, cela implique que la complexité de chiffrement égale à O(n^3).
  Et la complexité de déchiffrement dépend de la formule c'=c*p$^-$=m*G'*p$^-$, donc sa complexité =O(n^2).
  on obtient alors une complexité qui égale au O(n^3)).

4.1.5 Les cryptosystèmes basé sur le plus court vecteur non nul

La sécurité du ces cryptosystèmes est basée sur le problème de trouver le plus court vecteur non nul d'un réseau, NTRU est l'un de ces cryptosystèmes.

4.1.5.1 NTRU

Ce cryptosystème a été inventé entre 1996 et 1998 par J.H Silverman, J. Ho-stein et J. Pipher , qui est basée sur le problème du plus court vecteur non nul d'un réseau [53, 71, 76, 94, 98, 99, 100, 101, 103], le nom NTRU en hommage à leur groupe de travail "Number Theorists Research Units". Ce cryptosystème permet de chiffrer, déchiffrer ainsi de signer des messages et des clés avec NSS.

- **Principe** On se place dans l'anneau R=Z[X]/(X^n-1), et on pose les deux polynôme F et G telle que :
  F=$\sum_{i=0}^{n-1}$ F$_i$ X^i
  G=$\sum_{i=0}^{n-1}$ G$_i$ X^i
  On a alors les deux oppérations suivantes :

  – Addition F+G=(F$_0$ + G$_0$,F$_1$ + G$_1$,...,F$_{n-1}$ + G$_{n-1}$)=$\sum_{i=0}^{N-1}$ (F$_i$+G$_i$)X^i)

– Produit F $\bigotimes$ G= H = $\sum_{i=0}^{N-1} H_i X^i$

avec H_k=$\sum_{i=0}^{k} F_i$*G_{k-i} + $\sum_{i=k+1}^{n-1}$ (F_i*G_{n+k-i}

Le résultat se trouve dans le tableau 4.2 [71] :

	X^0=1	X^1	...	X^k	...	X^{N-1}
	f_0g_0	f_1g_1	...	f_kg_k	...	$f_{N-1}g_{n-1}$
+	f_1g_{n-1}	f_1g_0	...	f_1g_{k-1}	...	f_1g_{n-2}
+	f_2g_{n-2}	f_2g_{N-1}	...	f_2g_{k-2}	...	f_2g_{n-3}
...	...	...	...	...	...	....
+	$f_{n-2}g_2$	$f_{n-2}g_3$	...	$f_{n-2}g_{k+2}$	...	$f_{n-2}g_1$
+	$f_{n-1}g_1$	$f_{N-1}g_2$	...	$f_{n-1}g_{k+1}$	...	$f_{n-1}g_0$

TABLE 4.2 – Le produit de deux polynômes de NTRU

1. **Génération de clés** On prend deux polynômes f et g de manière aléatoire, qui doivent satisfaire deux conditions : Il doit avoir un inverse modulo q et un inverse modulo p, où q est un modulo important (comme 128 ou 256) et p est un petit modulo (comme 3, par exemple). On note ces inverses respectivement $f^{-1}{}_q$ et $f^{-1}{}_p$, avec :
   $f^{-1}{}_q \times f \equiv 1 \bmod q$
   $f^{-1}{}_p \times f \equiv 1 \bmod p$
   Enfin, l'algorithme calcule la clé publique h : $h \equiv f^{-1}{}_q \times g \bmod q$, ainsi la clé privée est (f,g)
2. **Chiffrement**
   Soit m le message en clair, sous la forme d'un polynôme Modulo q, l'algorithme génère le polynômes aléatoire r, puis calcule : e≡ m +pr $\bigotimes$h mod q qui représente le message chiffré.
3. **Déchiffrement** Ici la clé privée est constitué du polynôme f, l'algorithme calcule le polynôme intermédiare a : a ≡ f$\bigotimes$ e ≡ f$\bigotimes$ (m +pr $\bigotimes$ h) ≡ f $\bigotimes$ m + pr $\bigotimes$ g mod q. Enfin, a $\bigotimes$ $f^{-1}{}_p \equiv$ (f$\bigotimes$ m +pr $\bigotimes$ g)$\bigotimes$ $f^{-1}{}_p$ $\bigotimes$ m mod q.

- **Compléxité**

Le calcul dirtect de F*G a une compléxité O(n^2) . La transformé de Fourier (FFT) requiert uniquement une complexité en O(n log n)[76] lorsque n est une puissance de 2.

4.2 Avantages et limites des cryptosystème étudiés

4.2.1 Les cryptosystèmes basés sur la factorisation

1. **Les avantages de factorisation**

   - Le grand avantage de ces cryptosystèmes est que le texte chiffré peut seulement être déchiffré si l'attaquant est capable de factoriser efficacement la clé publique *n*.
   - Ces cryptosystèmes sont éventuellement des méthodes de chiffrement assez sûres.
   - En particulier l'algorithme RSA, est encore utilisable, par contre, pour avoir une sécurité optimale il ne faut plus utiliser des clés RSA inférieures à 1024 bits, pour des données sensibles, l'ANSSI et le NIST recommandaient déjà l'utilisation de clés de 2048 bits en 2010.
   - Particulièrement, le cryptosystème de Guillou quisquater ; il est plus résistant aux attaques que RSA, tel qu'il permet d'authentifier une carte à puce, en moins d'une seconde, sans partager de clés, et il est 40 fois plus rapide qu'un système utilisant RSA, il offre à la fois un service d'authentification et de signature.

2. **Les limites de factorisation**
   - La méthode de cryptage et décryptage pour la factorisation est longue en termes de calculs.
   - Attaque par factorisation :
     Cet attaque consiste à trouver la clé privée correspondante à la clé publique attaquée, en factorisant le *n* en deux nombres premiers *p* et *q* et trouver *d* en appliquant les fonctions convenables. Cependant cette technique est très fastidieuse lorsque n est très grand.
     Il existe un algorithme de factorisation baptisé QS (Crible Quadratique) de Pomerance qui est utilisé dans la mesure où le nombre à factoriser n'est pas trop grand [27, 10].

   - La signature RSA est universellement falsifiable sous des attaques à messages choisis.
     Dans un système RSA, disposant de deux signatures de deux messages différents (signatures produites avec la même clé privée), l'attaquant peut facilement produire une autre signature valide, sans posséder la clé privée. soit s_1=47 la signature du message m_1=5 avec la clé privée (d=37,n=77).
     s_2=28 la signature du message m_2=7 avec la clé privée (d=37,n=77).
     La signature du message m= $m_1 * m_2$=5 *7=35. On a $s_1 * s_2$= ($m_1{}^d$ mod n) *($m_2{}^d$ mod n)=47 *28 mod 77=1316 mod 77=7. $(m_1 * m_2)^d$ mod n =35^{37} mod 77=7. Ceci prouve que le produit des signatures des deux messages (réalisés avec la même clé privée) est égal à la signature du produit des deux messages, ce qui permet de créer des signatures valides sans posséder la clé privée. [80, 90]
   - En outre, le cryptosystème de Goldwasser n'est pas efficace en termes de stockage, de fait que la taille de texte chiffré est deux fois plus long que le texte original. En effet, la sécurité

sémantique ne considère que le cas d'un adversaire "passif" qui observe des textes chiffrés, et dans ce cas l'attaquant utilise l'attaque à texte chiffré choisi pour casser le système [61, 62].

– Notamment, l'un désavantage de Rabin dû à un non déterminisme, c'est que cet algorithme n'est pas injectif, pour un message chiffré reçu il y a quatre messages clairs donc la nécessité de faire un choix entre ces quatre possibilités [45]. Cela peut être surmonté en rajoutant une redondance aux textes clairs avant le chiffrement (par exemple, on peut ajouter un certain nombre avant le premier bit du message).Donc seul un des 4 messages aura un sens donc il n'y aura pas d'ambiguïté. Si aucun de ces messages ne commence par le bit ajouté, le message sera rejeté.

4.2.2 Les cryptosystèmes basé sur le logarithme discret

1. **Les avantages de logarithme discret**
   – Les exponentiations modulaires utilisées dans ces systèmes doivent être choisies avec une taille assez grande de manière à empêcher une recherche par l'intermédiaire d'un algorithme [47].
   – Particuliérement les avantages des courbes elliptiques sont [120] :
     – Ces dernières sont définies sur des groupes peu connus comparativement à (Z/pZ).
     – Les calculs sont à priori plus rapide et requièrent moins de mémoire.
     – Offre une taille réduite de la clé pour un même degré de sécurité que RSA (164 bits %1024).
     – Utilise des opérations mathématiques simples dans la génération des clés (+,-, x,/).
     – La taille des groupes de points est limitée mais suffisamment élevée.
2. **Les limites de Logarithme discret**
   – Un message chiffré par ses cryptosystèmes est plus long que le message clair.
   – La sécurité du système repose sur le problème du logarithme discret et le problème de Diffie Hellman car il se base sur ce principe afin de distribuer les clés. Ce qui nous donne une difficulté de calcul [47].
   – Afin d'assurer la sécurité du système d'Elgamal, l'expéditeur doit utiliser une nouvelle clé publique pour tout nouveau chiffrement. En effet, si l'expéditeur utilise deux fois la même clé pour chiffrer deux messages différents alors il suffit de connaître un des deux messages pour trouver l'autre[34].
   – Le seul inconvénient des courbes elliptiques est que leurs théorie est complexe et récente [120].

4.2.3 Les cryptosystèmes basés sur le problème de sac à dos

Parmi ces crtptosystèmes nous citrons, Merkel Hellman :

1. **Avantages**

   Si la suite utilisée est super croissante, alors le problème est facile et résolvable en un temps polynomial. On dit que le sac est super croissant. Cela rend la vitesse de chiffrement plus de 100 fois supérieure à RSA ; lié à un problème prouvé difficile [18].

2. **Limites**

   - Il est clair que plus la clé est longue, plus le message sera difficile à décrypter. En effet, si ce n'était pas le cas, on pourrait attaquer le texte chiffré par une analyse des fréquences, en faisant appel à l'attaque exhaustive, car chaque lettre serait chiffrée par le même nombre, le système est alors vulnérable à une attaque à l'aide d'une analyse de fréquence. Il convient donc de choisir des blocs de chiffrement de longueur inférieure à celle de la clé.
   - Malgré les modifications apportées à l'algorithme, à la suite de ces découvertes, l'outil qui a permis d'attaquer ce système est la réduction de réseau, tel que le principal algorithme de réduction de réseau, LLL permet de déchiffrer systématiquement le message chiffré. C'est pourquoi le chiffrement de Merkel Hellman n'est pas utilisé de nos jours.
   - Ce cryptosystème de Merkel Helman a été démontré comme vulnérable par Adi Shamir. Cependant, contrairement à RSA, il est à sens unique (n'admet pas de la signature), c'est-à-dire que la clé publique est utilisée uniquement pour le chiffrement, et la clé privée uniquement pour le déchiffrement. Il ne peut donc pas être utilisé pour un protocole d'authentification.

4.2.4 Les cryptosystèmes basés sur la théorie des codes

McEliece est l'un de ces cryptosystèmes :

1. **Avantages**
   - Généralement, les codes de Goppa sont considérés comme des bons codes linéaires puisqu'ils permettent de corriger plusieurs erreurs.
   - La sécurité repose sur deux notions : le problème de décodage des codes et ainsi la difficulté de déterminer lequel code utilisé parmi les codes de Goppa et les codes linéaires, cela rend la cryptanalyse est difficile.
2. **Limites**

   cet algorithme n'est pas utilisé en pratique car :
   - Les clés publiques et privées sont de grandes matrices, ce qui constitue un des plus grands désavantages de ce chiffre [105].

- Le texte chiffré est plus grand à celle du texte d'origine.
- Il y a eu des tentatives de cryptanalyse sur le cryptosystème de McEliece, mais sans succès.

4.2.5 Les cryptosystèmes basés sur le problème de plus court vecteur non nul

Dans cette classe nous prenons NTRU comme exemple :

1. **Avantages**

   - Sa rapidité et son efficacité en chiffrement et en déchiffrement, quoique ce soit au niveau d'espace mémoire et de calcul.
   - Sa grande force réside dans sa cryptanalyse ; contrairement aux algorithmes basés sur la factorisation tels RSA, qui peut-être mis à mal par des adversaires utilisant la factorisation du nombre chiffré en deux nombres premiers p et q, et cela par usage des calculateurs quantiques. Notamment, NTRU qui se base sur un problème complètement différent, ne possède pas encore de solution de cryptanalyse par ce type de calculateurs.
   - La sécurité de NTRU est liée à la difficulté de trouver le plus petit vecteur (SPV) dans un réseau euclidien de grande dimension.
     Cette caractéristique est intéressante puisqu'elle s'avère résistante aux attaques basées sur des ordinateurs quantiques, ce qui n'est pas le cas des algorithmes RSA ou ECC basés sur la factorisation d'entiers de grande taille.
   - NTRU reste toujours avantageux en termes de performance lorsqu'on le compare à RSA, en effet, lorsque la taille des clés augmente de n bits, le taux d'opérations par seconde décroît en n^3 pour l'algorithme RSA alors qu'il ne décroît qu'en n^2 pour NTRU. Ainsi, le niveau de performance de NTRU augmente avec le niveau de sécurité, ce qui est un atout remarquable [114].

2. **Limites**

   - Le décryptage peut ne pas fonctionner dans certains cas, Ceci est dû au fait qu'une équivalence modulo q peut être fausse modulo p et inversement, par exemple p=5, q=16, on prend x=17, x mod p= 17 mod 5=2 $\neq$ x mod q = 17 mod 16 =1.
   - La cryptanalyse de NTRU s'appuie sur l'algorithme LLL, en manipulant des polynômes dans Z[X]/(X^N-1) et en particulier en décomposant la clé publique h sous la forme d'un produit de deux polynômes f et g, on arrive à utiliser cet algorithme pour trouver une base et obtenir le plus petit vecteur. Cela permet de retrouver les polynômes f et g utilisés pour déchiffrer le message.

4.3 Conclusion

Dans ce chapitre, nous avons élaboré une étude des cryptosystèmes de chiffrement asymétrique structurée sous forme d'une taxonomie en plusieurs classes. Nous les avons classifiés le problème utilisé, ainsi nous avons déterminé les avantages et les limites de chaque classe.

CHAPITRE 5

ETUDE COMPARATIVE POUR LE CADRE DES RÉSEAUX DE CAPTEURS SANS FIL

Etablir une communication sécurisée est utile pour la majorité des applications des RCSF, le problème major qui se pose est comment établir des clés cryptographiques entre les nœuds capteur, tout en respectant les ressources limitées en mémoire avec une préservation des tailles de clés petites, ainsi une vitesse de chiffrement et déchiffrement rapide et une conservation d'énergie afin d'assurer la sécurisation des communications. En outre, vu les contraintes de limitation de ressources des nœud capteur, il ne sera inutile d'intégrer des cryptosystèmes si la gestion des clés est faible ainsi si les performances ne sont pas respectées. En conséquence de ces raisons, nous avons étudié les différents cryptosystèmes asymétriques dans le chapitre précédent, et dans ce chapitre nous les comparons selon des critères prédéfinis afin de déterminer lequel est le plus approprié pour l'applicabilité dans les RCSF.

5.1 Comparaison en termes de services de sécurité assurés

Maintenant que les hypothèses calculatoires sont posées pour chaque cryptosystèmes, nous nous intéressons à la façon dont elles peuvent être utilisées pour prouver leur sécurité. Nous étudions alors les divers services de sécurité assurés par chaque cryptosystèmes dans lesquels l'adversaire ne peut pas se placer pour mener son attaque, tels que : la confidentialité, l'authentification, l'intégrité de données et la non répudiation. Nous présentons sur le tableau 5.1 une comparaison globale. La comparaison est faite selon les points de vu suivants :

- **La confidentialité :** Avec les cryptosystèmes asymétriques, les clés de chiffrement et déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. La clé de chiffrement est publique tandis que celle de déchiffrement est privée. Pour un message chiffré, n'importe qui ne peut le chiffrer un message, sauf le propriétaire de la clé privée correspondante. Tous les cryptosystèmes étudiés adoptent ce mécanisme et ainsi assurent le service de confidentialité.
- **L'authentification, l'intégrité et la non-répudiation :** Les services d'authentification, l'in-

tégrité de données et la non-répudiation doivent être assurés par le mécanisme de signature numérique. Certains cryptosystèmes asymétriques n'assurent que le chiffrement tel que Paillier, Rabin Michael et Merkel-Hellman. D'autres sont utilisables à la fois pour le chiffrement et pour la signature tels que : RSA [77], Guillou-Quisquater [144], Goldwasser-Micali [143], ElGamal [65, 79, 135], Courbe Elliptique [123], McEliece [124] et NTRU [101].

	Les services de sécurité			
cryptosystème	La confidentialité	L'authentification	L'intégrité	La non-répudiation
RSA	✓	✓	✓	✓
Guilliou Quisquater	✓	✓	✓	✓
Pallier Pascal	✓	×	×	×
Golswasser Micali	✓	✓	✓	✓
Rabin	✓	×	×	×
ElGamal	✓	✓	✓	✓
Courbe Elliptique	✓	✓	✓	✓
Merkel Hellman	✓	×	×	×
McEliece	✓	✓	✓	✓
NTRU	✓	✓	✓	✓

TABLE 5.1 – Comparaison des cryptosystèmes asymétrique selon les services de sécurité assurés

5.2 Comparaison en termes de rapidité et taille des clés

En vue des résultats obtenus dans l'étude qui est faite dans le chapitre précédent, nous proposons d'étendre notre synthèse en s'appuyant sur le critère de complexité afin d'estimer la rapidité de calcul et la taille de clés. Cette étude, va nous permettre de voir de près les cryptosystèmes les plus appropriés aux contraintes matérielles des RCSF. Le tableau 5.2 montre une telle comparaison suivant certains critères qui sont :

- **La complexité algorithmique :** nous calculons le nombre d'opérations élémentaires pour chaque cryptosystème.
- **La rapidité de chiffrement/déchiffrement :** c'est la vitesse des calculs de chiffrement et déchiffrement qui dépend de nombre d'opérations exécutées pour un cryptosystème donné.
- **Taille des clés utilisées :** représente le nombre de bits de la clé utilisée.

La classe	Algorithme	Complexité	Rapidité	Taille des clés
Factorisation	RSA	O(log n)^3	Très lente	1024-2048
	Guillou Quisquater	O(log n)^3	Très lente	1024
	Pallier Pascal	O(log n)^3	Très lente	1024
	Golswasser Micali	O(log n)^3	Très lente	1024
	Rabin	O(log n)^2	Très lente	1024
Logarithme discret	El Gamal	O(log n)^3	Très lente	1024-2048
	Courbe Elliptique	$2^{((logn)\frac{1}{2}(log(logn))\frac{1}{2})}$	Très rapide	192
Sac à dos	Merkel Hellman	O(e^n)	Très lente	Grande taille
Codes correcteurs	McEliece	O(n^3)	rapide	n=1024, t=50, k>=524
Plus court vecteur non nul	NTRU	O(n$*$log(n))	rapide	251-503

TABLE 5.2 – comparaison des algorithmes asymétrique

5.3 Consommation d'énergie

L'énergie des capteurs doit être utilisée d'une manière optimale afin de maximiser la durée de vie du réseau. Dans cette section, nous décrirons la problématique de la consommation d'énergie dans les RCSF, en appliquant la cryptographie asymétrique[12, 13]. L'émission d'un signal est caractérisée par sa puissance ; quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur. Cette contrainte est prise en priorité dans notre étude, car les clés publiques doivent être diffusées sur le réseau. En effet, l'énergie consommée en envoyant une donnée de n bits vers un récepteur se trouvant à d mètres est estimée comme suit d'aprés Heinzelman et al : E(n,d)=n*(Eelec+Eamp*d^2) [142], tels que Eelec et Eamp représentent respectivement l'énergie de transmission électronique et d'amplification. Nous supposons qu'on dispose des nœuds capteurs de type Xm2110 [4], dont leurs caractéristiques sont présentées dans le tableau 5.3.

f=2,4 Ghz =2,4*10^9 hz $\rightarrow \tau = \frac{1}{f}=\frac{1}{2,4} * 10^{-9}$ =0,42*10^{-9} secondes.

capteur	fréquence	processeur	Ram	énergie	vitesse d'exécution
Xm2110	2,4Ghz	Atmel	8 Kbyts	10 joule	2,4*10^9 inst/scd

TABLE 5.3 – Les caractéristiques de capteur Xm2110

Nous supposons une telle opération dont son temps d'exécution= 10^{-9} secondes consomme "0,1" joule d'énergie. Dans la suite nous calculons le temps d'exécution des opérations de chiffrement et

de déchiffrement de chaque cryptosystème afin d'estimer l'énergie consommée en effectuant ces opérations et celle de communication des clés publiques. Pour cela, nous utilisons les mêmes paramètres de chiffrement pour tous les cryptosystemes se trouvant dans la même classe, afin de pouvoir les comparer et ainsi de simplifier les calculs.

- **La factorisation**
  Soit : p=23, q=19, n=p*q=437, φ(n)=396, e=29, d=41, m=22 =10110
  Nous avons e*d-kφ(n)=1 $\longrightarrow$ k=$\frac{d*e-1}{\varphi(n)}$=3.

Comme nous avons besoin aussi de calculer le CPI (c.à.d le nombre de cycle pour une instruction de type i) tel que sa formule est la suivante :
CPI_{moy}= $\sum$(CPI_i *NI_i)/ NI
NI_i : le nombre d'instructions de type i.
NI : le nombre d'instructions globale du programme.
Nous supposons le même CPI_i=1 pour chaque type d'instruction. C'est à dire un seul cycle pour chaque instruction.
Donc, CPI_{moy}=$\sum$(CPI_i / NI)=1
Texc=CPI_{moy}*Ni*τ=Ni*τ=Ni*0,42*10^{-9}scd.

1. **Evaluation de RSA** :

Opération	Compléxité
n = p$*$q	1
φ(n)=(p-1)$*$(q-1)	3
Calculer d tel que e$*$d mod φ(n)=1	5$*$ k=15
c= m^e mod n	e=29
m = c^d mod n	d=41
nombre d'opérations	nbr-op= 89

TABLE 5.4 – Evaluation de RSA

Texc-RSA=nbr-op*τ=89*0.42*10^{-9}=37.38*10^{-9} scd.
E-RSA-calc=3,74 joule.
La clé publique est (e,n) codifié sur (1024,512)=(2^{10},2^9),nombre de bites=10,6 donc l'énergie d'émission E-RSA-emis=10,6*0,3=3,18 joule.
Donc l'énergie consommée est 3,74+3,18=6,92 joule.

2. **Evaluation de Guillou Quisquater** :

Soit Ja=6 donc Ja^{-}=73 et k'=1 , e=11,v=29, s=41

Opération	Compléxité
n = p$*$q	1
φ(n)=(p-1)$*$(q-1)	3
Calculer s tel que v$*$s mod φ(n)=1	5$*$ k=15
Calculer Sa tel que Sa= Ja^{-s} mod n	5$*$ k'+s=46
x=m^{v} mod n	v=29
y=m$*$ Sa^{e} mod n	e+1=12
Ja^{e} $*$ y^{v} mod n	e+v=40
nombre d'opérations	nbr-op= 146

TABLE 5.5 – Evaluation de Guillou

Texc-Guillou=nbr-op*τ=146*0,42=61,32*10^{-9} scd.
E-Guillou-calc=6,13joule.
La clé publique est (e,n), donc l'énergie d'émission E-Quillou-emis=10,6*0,3=3,18 joule.
Donc l'énergie consommée est 6,13+3,18=9,31 joule.

3. **Evaluation de Paillier** :

Opération	Compléxité
n = p$*$q	1
φ(n)=(p-1)$*$(q-1)	3
Calculer r^{-} tel que e$*$d mod φ(n)=1	5$*$ k=15
c= (1+ n)^{m} * r^{n} mod n^{2}	m+n=459
m=$\frac{(c*r^{-n}modn^{2})-1}{n}$	n+4=441
nombre d'opération	nbr-op= 919

TABLE 5.6 – Evaluation de Paillier

Texc-Paillier=nbr-op*τ=919*0,42=386*10^{-9} scd. E-Paillier-calc=38,6 joule.
La clé publique est (r,n), donc l'énergie d'émission E-Paillier-emis=10,6*0,3=3,18 joule.
Donc l'énergie consommée est 38,6+3,18=41,78 joule.

4. **Evaluation de Goldwasser Micali** :

Soit le message =22=10110 , donc la taille t =5.
Texc-Goldwasser=nbr-op*τ=1000*0,42=420*10^{-9} scd. cosom-Goldwasser=42 joule.
La clé publique est (z,n), donc l'énergie d'émission E=10,6*0,3=3,18 joule
Donc l'énergie consommée est 42+3,18=45,18 joule.

Opération	Compléxité
n = p∗q	1
résidu quadratique	2*n=874
le symbole de Jacobi	p+q=42
chiffrement de m	3*t=15
déchiffrement de c	p*t=69
nombre d'opérations	nbr-op=1000

TABLE 5.7 – Evaluation de Goldwasser

5. **Evaluation de RABIN** :

Opération	Compléxité
n = p∗q	1
c=m^2mod n	2
$c^{\frac{p+1}{4}}$ mod p	6
$c^{\frac{q+1}{4}}$ mod p	5
p^-	30
q^-	70
$m_i = \pm p * p^- * q_1 \pm q * q^- * p_1$ mod n	20
nombre d'opérations	nbr-op= 134

TABLE 5.8 – Evaluation de Rabin

Texc-Rabin=nbr-op*τ=134*0,42=56,28*10^{-9} scd. E-rabin-calc=5,63 joule.
La clé publique est (n), donc l'énergie d'émission E=10*0,3=3 joule Donc l'énergie consommée est 5,63+3=8,63 joule.

6. **Evaluation d'El-Gamal** :

Soit p=23, g=19, s=21, k=8

Opération	Compléxité
Calculer la clé publique b=g^s mod p	s=21
c1=g^k mod p, et c2=m*b^k mod p	k=8
r_1=$c_1{}^s$ mod p	s=21
m=c_2/r_1	1
nombre d'opérations	nbr-op= 50

TABLE 5.9 – Evaluation d'El-Gamal

Texc-Elgamal=nbr-op*τ=50*0,42=21*10^{-9} scd.
cosom-Elgamal=2,1 joule.
La clé publique est (b), donc l'énergie d'émission E=10*0,3=3 joule Donc l'énergie consommée est 2,1+3=4,1 joule.

7. **Evaluation des courbes elliptique** :

Opération	Compléxité
x=m$*$k+j avec k=20	2
calculer y^2	2
cherchons si y^2 est un caré d'un nombre modulo p	4
cherchons si y^2 est un caré d'un nombre modulo p	2
m=$\frac{x-j}{k}$	2
nombre d'opérations	nbr-op= 12

TABLE 5.10 – Evaluation d'ECC

Texc-ECC=nbr-op*τ=12*0,42=5*10^{-9} scd.
cosom-ECC=0,5 joule.
La clé publique est le point(x,y) codifiée sur 2^6, donc l'énergie d'émission E=6*0,3=1,8 joule Donc l'énergie consommée est 0,5+1,8=2,3 joule.

8. **Evaluation de Merkle-Hellman** :

Opération	Compléxité
calculer l'inverse de e	5
construire la suite publique	n*e=35*9=315
chiffrement de m	n-1=34
déchiffrement de c	n-1=34
nombre d'opérations	nbr-op=400

TABLE 5.11 – Evaluation de Merkle-Hellman

Texc-Merkel-Hilman=nbr-op*τ=400*0,42=168*10^{-9} scd.
cosom-ECC=16,8 joule.
La clé publique est (la suite non supercroissante s' qui est constitué de n éléments), donc l'énergie d'émission E=49*0,3=14,7 joule.
Donc l'énergie consommée est 16,8 +14,7=31,5 joule.

9. **Evaluation de McEliece** :
Soit n=3,k=4 t=3.

Opération	Compléxité
calcule de la matrice inversible S	n^2=9
calcule de l'invrese de la matrice de permutation P	k^2 =16
calcule de la matrice publique G'=SGP	n^2*k^2 =25
calcul de M*G'	k*(2*n-1) =20
l'ajout d'un code e : c=c'+e	1
extraction de code : a=c*p^{-1}	1
extraction de message clair	1
nombre d'opérations	nbr-op= 72

TABLE 5.12 – Evaluation de McEliece

Texc-McEliece=nbr-op*τ=72*0,42=30,24*10^{-9} scd.
cosom-McEliece=3,024 joule.
La clé publique est (G' : codé sur 2^{20}), donc l'énergie d'émission E=18*0,3=5,4 joule Donc l'énergie consommée est 3,024+5,4=8,42 joule.

10. **Evaluation de NTRU** :
Soit n=3,k=4 t=3.

Opération	Compléxité
calcule de l'inverse de f modulo p et q	4
calcule de message chiffré	6
déchiffrement	6
nombre d'opérations	nbr-op= 16

TABLE 5.13 – Evaluation de NTRU

Texc-NTRU=nbr-op*τ=16*0,42=6,72*10^{-9} scd.
cosom-NTRU=0,672 joule.
La clé publique est (h : codé sur 2^7), donc l'énergie d'émission E=7*0,3=2,1 joule Donc l'énergie consommée est 0,672+2,1=2,77 joule.

Le tableau(5.14)récapitulatif des estimations des cryptosystèmes asymétrique étudiés.

Nous constatons que les cryptosystèmes qui s'exécutent leur opération en un temps d'exécution très lente sont celles de la classe factorisation, dû au fait que la factorisation des nombres nécessite largement de calcul, de même pour le cryptosystème Merkel Hellman tel que son temps d'exécution égale à 168 *10^{-9} secondes.
D'une autre part, l'énergie à consommer dépend de l'énergie de calcul et l'énergie de transfert de clés publiques, cela se prouve de fait que leurs haute capacité de calculs qui préserve un temps très

	Les performances			
cryptosystème	Temps d'exécution(10^{-9}scd)	Enérgie du calcul(joule)	Energie du transfer	L'énérgie consommé
RSA	37,38	3,74	3,18	6,92
Guilliou Quisquater	61,32	6,13	3,18	9,31
Pallier Pascal	386	38,6	3,18	41,78
Golswasser Micali	420	42	3,18	45,18
Rabin	56,28	5,63	3	8,63
ElGamal	21	2,1	3	5,1
Courbe Elliptique	5	0,5	1,8	2,3
Merkel Hellman	168	16,8	14,7	31,5
McEliece	30,24	3,024	5,4	8,42
NTRU	6,72	0,672	2,1	2,77

TABLE 5.14 – La consommation d'enérgie des cryptosystèmes asymétrique

lent et à leurs grandeur de clés, donc cela nous provoque une consommation d'une grande quantité d'énergie.

Par contre, pour la classe logarithme discret, le temps d'exécution des courbes elliptiques est meilleur que celui de cryptosystème d'Elgamal qui mène le cryptosystème d'ECC à consommer très peu d'énergie par rapport à Elgamal. En outre le cryptosystème de NTRU nécessite des valeurs proches à celle d'ECC.

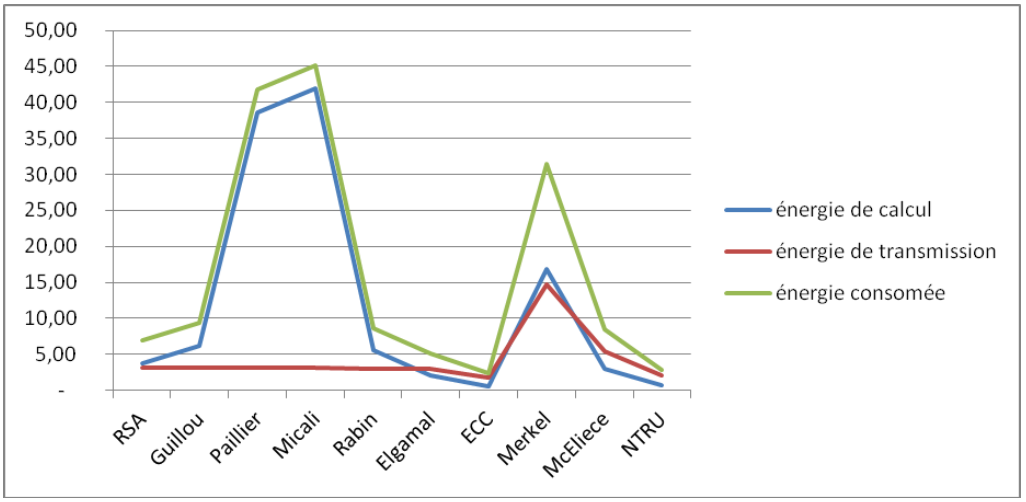

FIGURE 5.1 – L'énergie consommée des cryptosystèmes asymétriques

5.4 Fréquence d'horloge

D'après le tableau 5.14, nous remarquons que le cryptosystème basé sur les courbes elliptiques (ECC) a un temps d'exécution meilleur par rapport à d'autres cryptosystèmes. Pour mettre l'accent sur les ressources matérielles, on va supposer que tous les cryptosystèmes étudiés doivent s'exécuter dans un temps d'exécution bien déterminé, soit T=5 $*10^{-9}$ secondes.
Qu'elle est la fréquence d'horloge que doit le processeur avoir pour exécuter une opération de chiffrement dans le temps déterminé T ?

Cryptosystème	RSA	Guillou	Paillier	Micali	Rabin	Elgamal	ECC	Merkel	McEliece	NTRU
Fréquence(Ghz)	17,8	29,2	183,5	200	26,8	10	2,4	80	14,5	3,23

TABLE 5.15 – Les fréquences d'horloge des cryptosystèmes asymétrique

En outre, puisque la puissance de calcul d'un processeur est très importante pour les réseaux de capteurs sans fil, le tableau 5.15 illustre les déférentes fréquence d'horloge des cryptosystèmes étudiés, et cela après avoir supposé que leurs opérations s'effectuent en temps égal à 5*10^{-9}.
Nous remarquons que le cryptosystème de Goldwasser Micali nécessite pour ces calculs une fréquence d'horloge qui égale à 200 GHZ, qui est une fréquence trop grande.
Et de même pour les autres algorithmes de la classe factorisation RSA, Paillier, Rabin, Guillou, ainsi Merkel Hellman, et cette grandeur est dû au fait que le nombre d'opérations exécutées par ces cryptosystèmes est énormément grandes, surtout pour la classe factorisation des nombres, et dans la classe de logarithme discret on trouve ECC nécessite une fréquence d'horloge=2,4 GHZ qui est de même égale à celle de nœud capteur proposé.

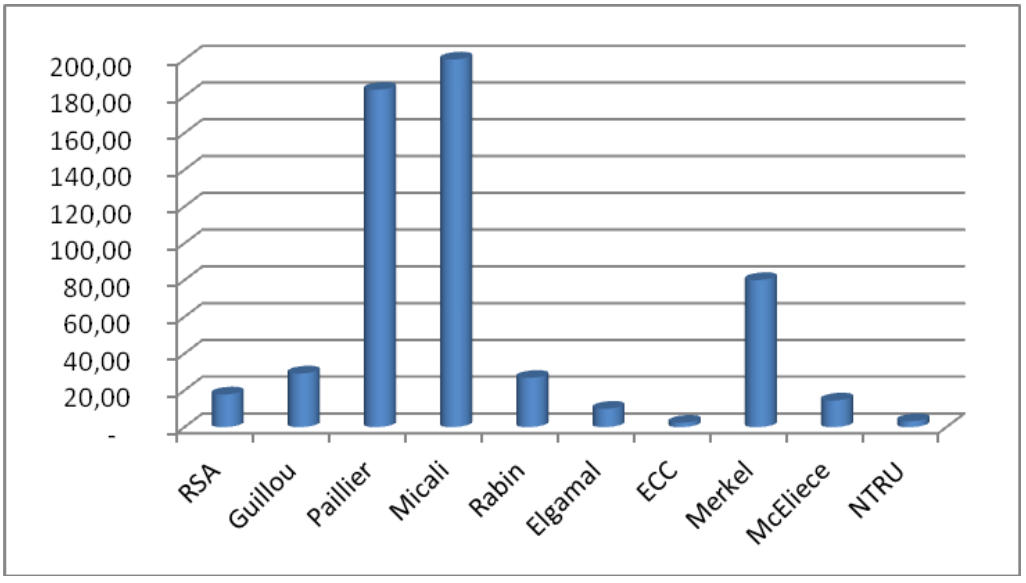

FIGURE 5.2 – La fréquence d'horloge des cryptosystèmes asymétriques

5.5 La capacité de la mémoire de stockage

cryptosystème	RSA	Guillou	Paillier	Micali	Rabin	Elgamal	ECC	Merkel	McEliece	NTRU
memoire(K bit)	24,25	24,25	24,25	24,25	16	16	0,125	1024	512	0,25

TABLE 5.16 – La taille de la memoire des cryptosystèmes asymétrique

D'autre part, parmi les contraintes des RCSF, c'est la contrainte de stockage qui est très limitée. Nous avons étudié pour chaque cryptosystème, la taille des clés publiques stockées ainsi que la capacité de stockage nécessaire.
D'après cette étude, nous avons pu résumer les résultats trouvés dans le tableau **??**, nous constatons que la taille des clés stockées est trop volumineuse dans la classe factorisation. Ce qui est causé par l'utilisation des clés publiques composées des deux grands nombres, tel que nous remarquons que les cryptosystèmes de Paillier, Goldwasser Micali et Guillou utilisent la même taille de clé qui égale à 1024 bits, de même pour RSA.
Néanmoins, cette clé est différente pour Rabin qui stocke uniquement le n, par contre les cryptosystèmes Merkel Hellman et McEliece dont leurs clés publiques sont de grande taille du fait que Merkel utilise des suites et McEliece utilise des matrices qui sont de taille importantes ce qui demande un espace de stockage trop grand par rapport à d'autres cryptosystèmes.
D'une autre part, les cryptosystèmes des courbes elliptiques et NTRU ne demandent pas trop d'espace de stockage.

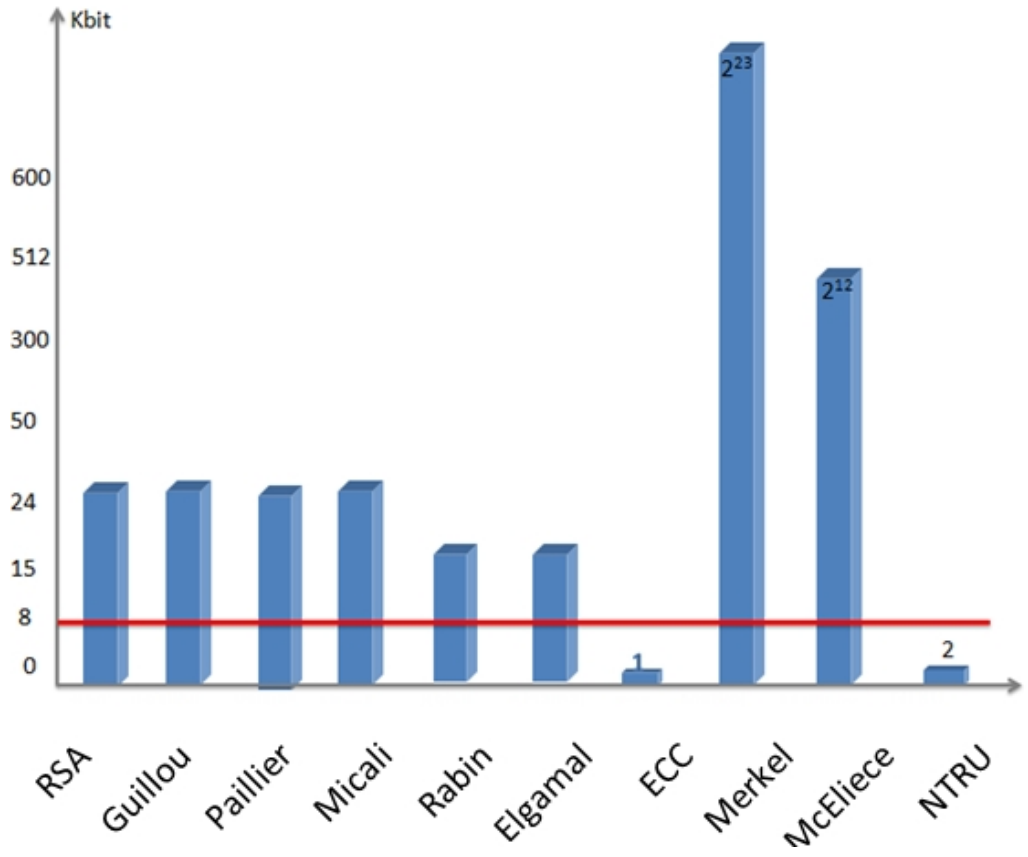

FIGURE 5.3 – La capacité mémoire des cryptosystèmes asymétriques

5.6 Conclusion

Dans ce chapitre de nombreux algorithmes proposés pour la cryptographie à clés publiques se sont révélés rapidement non sûrs, ou non réalisables sur le plan pratique. Tous les algorithmes actuels présentent l'inconvénient d'être bien plus lents que les algorithmes à clés secrètes ; de ce fait, ils sont souvent utilisés non pour chiffrer directement des données, mais pour chiffrer une clé de session secrète.

D'autre part RSA à toujours été utilisé, elle a été créée en même temps que celle de Merkel-Hellman et demande des clés plus petites pour une résistance comparable, mais le cryptosystème de Merkel-Hellman reste plus simple que RSA, en terme de calcule et de rapidité. En outre, certains algorithmes asymétriques ne sont adaptés qu'au chiffrement tel que (Merkel-Hellman, Rabin, etc.), tandis que d'autres ne permettent que la signature, et autres algorithmes sont utilisables à la fois pour le chiffrement et pour la signature tels que : RSA, ElGamal, ECC, etc. L'inadaptation de la cryptographie asymétrique a conduit les recherches afin de contribuer à une solution dans l'avenir. Des méthodes de cryptographie asymétrique à faible coût comme ECC ont un avenir prometteur pour sécuriser les RCSF et méritent des études plus approfondies.

CONCLUSION GÉNÉRALE

Les RCSF sont une nouvelle technologie qui a émergé après les grands progrès technologiques concernant le développement des capteurs intelligents. La conception et la maintenance de ce type de réseaux constituent un domaine de recherche très actif. Ce type de réseau a pour but la collecte de données de l'environnement et leur diffusion vers une station de base pour le traitement. Les éléments du réseau ont de petites dimensions et de sévères contraintes de ressources. Les RCSF constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication, mais avec toutefois des contraintes spécifiques. Parmi les sérieux problèmes posés à l'heure actuelle dans ce type de réseaux est la sécurité qui a fait l'objet de ce mémoire.

En premier lieu, nous avons donné une vue globale sur les RCSF en termes d'architecture, caractéristiques et domaines d'application. Ensuite, nous avons présenté les notions élémentaires de cryptographie, dans lesquelles nous avons présenté ses deux grandes familles de systèmes de chiffrement : symétrique et asymétrique. Par la suite, nous avons abordé les aspects de sécurité liés aux RCSF en termes d'attaques et solutions, et enfin nous avons proposé une classification avec une étude approfondie des systèmes de chiffrements à clés publiques. Pour conclure, nous tenons à préciser que notre contribution comporte deux parties essentielles :

1. Nous avons proposé une taxonomie des systèmes de chiffrement à clés publiques que nous considérons originale d'après notre lecture. Nous avons étudié en détail chaque système de chiffrement en termes d'avantages, inconvénients et complexité.
2. Nous avons également étendu notre synthèse pour le cadre des RCSF afin de comparer toutes les solutions en termes de matériels requis, pour estimer la pertinence de chacune vis-à-vis la technologie actuelle qu'on dispose.

Comme perspective, nous envisageons de raffiner notre étude à travers des simulations en comparant l'ensemble des systèmes de chiffrement asymétriques. Ceci va permettre de mieux estimer le temps nécessaire pour le chiffrement/déchiffrement et l'overhead de transmission des clés. Enfin, nous envisageons aussi d'améliorer et publier cette taxonomie sous forme d'un *Survey*.

Bibliographie

[1] M.KHANOUCHE "Le traitement du problème de la couverture dans les réseaux de capteurs sans fil", Thèse de Magistère, Université de béjaia 2008.

[2] N.SAIDANI, M.SADOU "Méthodes Géométriques Pour Le Traitement Du Problème De Couverture Dans Les Réseaux De Capteurs Sans Fil" Mémoire de master2 à l'université de Bejaia 2011.

[3] K.DAHOUMANE, K.NAIT ATMANE, "Conception et Implémentation d'un système de Gestion de Certificats pour les Réseaux de Capteurs Sans Fil", Mémoire d'ingénieur à l'université de Bejaia 2011.

[4] N.GUEDDOUDJ, S.GUELLAL "Approche d'agrégation de données pour la conservation d'énergie dans les réseaux de capteurs sans fil", mémoire d'ingénieur à l'université de Bejaia 2009.

[5] L.KHELLADI & N.BADACHE , "Les réseaux de capteurs,état de l'art" 2004.

[6] N.Lasla "La gestion de clés dans les réseaux de capteurs sans fil" ,thèse de magister à l'INI 2007.

[7] M.Messai "Sécurité dans les Réseaux de Capteurs Sans-Fil", thèse de magister à l'univertsité de Bejaia 2008.

[8] R.Kacimi "Techniques de conservation d'energie pour les réseaux de capteurs sans fil", thèse de Doctorat a l'université de Toulouse 2009.

[9] INSA de Lyon, Marine Minier, "Sécurité dans les réseaux ad hoc(2)".

[10] Ainigmatias Cruptos,"Attaque de RSA par fractions continues", Acrypta

[11] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital, Signatures and Public-Key Cryptosystems".

[12] G.Gaubatz,"Public Key Cryptography in Sensor Networks-Revisited", ESAS : 1st European Wksp, Security in Ad-Hoc and Sensor Networks, 2004.

[13] Krzysztof Piotrowski, Peter Langendoerfer and Steffen Peter, "How Public Key Cryptography Inuences Wireless Sensor Node Lifetime", 2006.

[14] A.Menezes,P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[15] S.ATHEMANI, protocole de sécurité pour RCSFs, mémoire de magistère, université de Batna 2010.

[16] M.Omar, Y.Challal, A.Bouabdallah, "Architecture de Certification Distribuée à base de Multi signature", Submitted to SAR SSI 2011.

[17] Hervé Schauer consultants, "Fonctionnement des PKI",1989.

[18] Damien Stehlé Istanbul,RSA,"Cryptosystèmes Sacs-à-dos et Réseaux Euclidiens",2005.

[19] Daniel and Robert "The static Diffi Hilman problem", 2005.

[20] Yannick Chevalier, "résolution de problèmes d'accessibilité pour la compilation et la validation de protocoles cryptographiques", Thèse Doctorat, à l'université Henri Poincaré Nancy, 2003.

[21] Une Introduction à la Cryptographie (news :fr.misc.cryptologie,1998).

[22] http ://www.hsc.fr/ressources/cours/pki/01_bases_crypto.pdf

[23] A.Bonnecaze & R.Rolland, "Arithmétique et Cryptographie, Institut de Mathématiques de Luminy (IML)".

[24] Ghislaine Labouret, "Introduction à la cryptologie, Novembre",1998.

[25] Pierre-Alain FOUQUE,École normale supérieure, "Initiation à la cryptographien, Factorisation, Log discret et Intégrité, Authentification".

[26] Jacques Stern, Louis Granboulan,Phong Nguyen,David Pointcheval, "Conception et preuves d'algorithmes cryptographiques, Cours de magistère M.M.F.A.I", 2004.

[27] Daniel Lerch Hostalot,"Attaque par factorisation contre RSA",2007.

[28] Nicolas DOUZIECH - Thomas JANNAUD-X2005,"Cryptologie,Attaque de clés RSA par la méthode de Wiener" ,2008.

[29] Marine Minier & INSA Lyon "Arithmétique pour la cryptographie".

[30] NGUYEN Tuong Lan - LIU Yi, "Cryptographie RSA, Introduction, Opérations, Attaques".

[31] Abderrahmane NITAJ,"Cles faibles pour le cryptosysteme RSA ", Université de Caen département de Mathématiques France Oujda, 26 Avril 2007.

[32] Y.CHALLAL. "Réseaux de capteurs sans fil", 2008.

[33] Steve Gury, Nicolas Rémond,"Cryptologie, El Gamal d'aprés Diffie-Hellman", 2004.

[34] Christophe Ritzenthaler ,"Chiffrement ElGamal et attaques sur le logarithme discret Option agregation", 12 December 2007.

[35] Sylvain Pasini, "Pourquoi les versions théoriques d'ElGamal et RSA ne sont pas sûres ?",SSC,Projet de semestre ,Mars 2005.

[36] Veronique Cortier,"Introduction à la cryptographie,Ecole des Mines".

[37] E.Brisson,"cryptographie,signature électronique",SGDN/DCSSI,laboratoire de cryptographie.

[38] C. Guillet & R. Yombi, "le knapsack et le chiffre de merkle hellman",20 mai 2009.

[39] ORANCI Sevan,POURROY Louis,E.Chanthery,"Cryptage et le problème du sac à dos",Compte rendu du T.I.P.E sur une Etude Bibliographique,2005-2006.

[40] F.DIMITRIOU,"l'application de la cryptologie en en matiére de sécurité des réseaux informatique", université de Lille2,2002.

[41] http : //www.futura-sciences.com/fr/news/t/high-tech-4/d/le-chiffrement-par-courbes-elliptiques-casse-a-109-bits_1363

[42] "La cryptographie à clé publique, chiffrement de Rabin".

[43] E.Bersson, "Cryptographie : Identification et Zero-Knowledge".

[44] "An Introduction to Cryptography",Network Associates, Inc, and its Affiliated Companies, 1990-2000.

[45] Zoubida Jadda et Patrice Parraud, "Mémento de Cryptologie", Ecoles Militaires de saint Cyr Coëtquidan.

[46] Complexité, Luc Brun , A partir de travaux de Habib Abdulrab(Insa de Rouen).

[47] http ://personnel.univ-reunion.fr/starento/cours_crypto7.pdf

[48] Roberto Amadio, "Une introduction à la cryptographie et aux protocoles cryptographiques", Master Université Paris Diderot (Paris7),2011 2012.

[49] http ://www.apprendre-en-ligne.net/crypto/rabin/index.html

[50] A.Zioui & N.Mouloua, "Courbe elliptique et leurs application en cryptographie" thèse,Université de béjaia 2007.

[51] http ://perso.univ-lr.fr/gbailly/cours/chapitre5.pdf

[52] Ayoub Otmani,"Cryptographie fondée sur la théorie des codes".

[53] http ://veille-techno.blogs.ec-nantes.fr/index.php/2012/01/06/cryptologie-ntru-number-theorists-r-us/

[54] http ://perso.univ-lr.fr/gbailly/cours/chapitre6.pdf

[55] Olivier Markowitch, "Cryptographoie et sécurité des systèmes informatiques".

[56] http ://www.picsi.org/parcours_39_175.html

[57] http ://www.vialo.net/themenreihe.p_c=Algorithme.

[58] Mines de Nancy,"Factorisation d'entiers",Pépites algorithmiques,mai 2009.

[59] http ://www.di.ens.fr/∼wwwgrecc/Rapports/rliens01/GRECC.html-C-BriPoVaYu00.

[60] http ://www.ceras-projet.org/index.php_id=3485.

[61] http ://fr.m.wikipedia.org/wiki/Cryptosyst%C3%A8me_ de_Goldwasser-Micali.

[62] http ://fr.m.wikipedia.org/wiki/IND-CCA.

[63] http ://www.picsi.org/fiche_143.html

[64] Philippe Hoogvorst, "introduction à la cryptographie", Télécom-Paristech.

[65] V.Meier, The ElGamal Cryptosystem Andreas, Juin 2005

[66] Ann Canteaut,"La cryptologie moderne".

[67] A.D. Wood and J.A. Stankovicn, "A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks".

[68] Frank Stajano and Ross J. Anderson. "The resurrecting duckling : security issues for ad-hoc wireless networks",1999.

[69] David Martins,Hervé Guynnet,"Etat de l'art sécurité dans les réseaux de capteurs sans fil", Submitted to SAR-SSI, 2008.

[70] Singh Simon, Histoire des codes secrets, Editions JC Lattès, 1999.

[71] Abderrahmane Nitaj, "La cryptographie du futur", Laboratoire de Mathématiques Nicolas Oresme,Université de Caen, France,(http ://www.math.unicaen.fr/∼nitaj).

[72] A. Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Pu-blishers, Boston, 1993.

[73] http ://www.ulb.ac.be/di/scsi/markowitch/crypto/Slides/goldwasserMicali.pdf

[74] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation.

[75] V.S. Miller, "Use of elliptic curves in cryptography", Crypto 85, 1985.

[76] Abderrahmane Nitaj,"Le cryptosysteme NTRU",Janvier 2003.

[77] Émeline Hufschmitt, "Signatures pour l'anonymat fondées sur les couplages et applications",version 1,25 Feb 2008.

[78] Christophe Giraud, "Attaques de crypto systèmes embarqués et contre-mesures associées", 26 octobre 2007.

[79] http ://en.wikipedia.org/wiki/ElGamal_signature_scheme

[80] Malika Izabachène, "L'anonymat dans les protocoles cryptographiques", Thèse du Doctorat de l'université Paris VII- Denis Diderot.

[81] Davide Alessio, "quelques questions de cryptographie : Anonymat révocable et une généralisation du chiffrement de Goldwasser-Micali", Thèse de Doctorat , Université de Rennes 1.

[82] http ://www.lastree.net/fragmentslog/fragments/anet_izmitli.pdf

[83] Y.CHALLAL,H.BETTAHAR,A.BOUABDALLAH ,"Cours1 : Les Réseaux de capteurs", Heudiasyc UMR CNRS 6599,Université de Technologie de Compiègne, France.

[84] Ivan Damgard,Mads Jurik and Jesper Buus Nielsen,"A Generalization of paillier's Public-key System with application to Electronic voting".

[85] Steven Galbraith, "The RSA and Rabin Cryptosystems", the book "Mathematics of Public Key Cryptography", by, available from http ://www.isg.rhul.ac.uk/sdg/crypto-book.

[86] Nicolas T. Courtois,"La sécurité des primitives cryptographiques basées sur des problèmes algèbriques multivariables :MQ,IP,Minrank, HFE",thèse de Doctorat de l'Université Paris 6, Soutenue le 25 Septembre 2001.

[87] David Pointcheval, "Le Chiffrement Asymétrique et la Sécurité Prouvée",2002.

[88] Guilhem Castagnos,"Quelques schémas de cryptographie asymétrique probabiliste",Thèse de Doctorat dirigée par François Arnault et Thierry Berger, 2006.

[89] Nirav Jobanputra, Vijayendra Kulkarni, Dinkar Rao, and Jerry Gao, Ph.D. and San Jose State University, "Emerging Security Technologies for Mobile User Accesses".

[90] Malika Izabachène and David Pointcheval,"New anonymity notions for identity-based-encryption",Ecole normale supérieure, France, 2008.

[91] Marc Joye and Pascal Paillier, "Fast generation of prime numbers on portable".

[92] http ://www.inetlab.net/certsrv/CryptoGraphie.htm

[93] Rolland Balzon Philippe,"Principaux algorithmes de cryptage",11 juillet 2002.

[94] Abderrahmane Nitaj,"Le cryptosysteme NTRU realites et perspectives", Université de Caen,Département de Mathématiques,France, Oujda, 27 Avril 2007.

[95] Jean-Guillaume Dumas,"Factorisation d'entiers cryptographique".

[96] Abderrahmane Nitaj, "Applications de l'algorithme LLL en cryptographie", Laboratoire de Mathématiques Nicolas Oresme, Université de Caen, France.

[97] Benny Chor and Ronald L.Rivest "A Knapsack-Type Public Key Cryptosystem Based on Arithmetic in Finite Fields".

[98] Marion Candau "Le cryptosystéme NTRU",14 mars 2011.

[99] Abderrahmane Nitaj,"NTRU et ses variantes, séecurité et applications".

[100] Jeff Hoffstein & Daniel Lieman & Jill Pipher & Joseph H. Silverman,"NTRU : a public kay cryposystem,NTRU Cryptosystems".

[101] Jeffrey Hoffdtein,Nick Howgrave-Graham,Jill Piper,Joseph H,Silverman,William Whyte "Ntru Sign : Digital Signature Using the NTRU Latiice".

[102] Yves GERARD, "Cryptographie et sécurité", Université Lyon1,2009.

[103] Damien STEHLE, "Trouver un vecteur le plus court dans un réseau euclidien" ,Université de Lyon.

[104] Springer-Verlag and Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes".

[105] C.Aguilar,P.Gaborit,F.Laguillaumie et A. Otmani,Université de Limoges, Université de Caen et INRIA.

[106] Jean-Marc Alliot1,"Cryptographie".

[107] "Paillier's Cryptosystem",

[108] Pascal Paillier and Moti Yung,"Self-Escrowed Public-Key Infrastructures".

[109] Ann Hibner KOBLITZ, Neal KOBLITZ and Alfred MENEZES, "Elliptic curve cryptographie :the serpentine course of a paradigm shift".

[110] boumbo.toonywood.org/xavier/maths/tipe.doc

[111] Sami Harari,"A Cryptosystem Using Error Correcting Codes and Correlations Laboratoire Modélisation et Signal",Université de Toulon.

[112] http ://www.les-mathematiques.net/phorum/read.php ?3,551643,551674

[113] http ://www.math.u-psud.fr/ montcouq/Enseignements/Codage/codeslin1.pdf

[114] http ://veille-techno.blogs.ec-nantes.fr/index.php/2012/01/06/cryptologie-ntru-number-theorists-r-us/

[115] H.Alatrista,S.Aliaga,K.Goua¨ých,J.Mathieu,"Implémentation de protocole sur une plateforme de réseaux de capteurs sans-fils",2008.

[116] Olivier Markowitch,"Sécurité des systèmes informatiques, Le chiffrement asymétrique".

[117] Pierre-Alain Fouque,"Le partage de clés cryptographiques :Théorie et Pratique" Thèse de Doctorat de l'université Paris7,2001.

[118] Nils Gura,Arun Patel,Arvinderpal Wander,Hans Eberle,Sheueling Chang Shantz,"Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs".

[119] "Announcing the advenced encryption standard(AES)",2001.

[120] André BOUMSO,"Méthode exploratoire de distribution des clé de cryptage pour les communications de groupe dans un réseau mobile ad hoc",Thèse à l'université du QUÉBEC à trois-rivières,2006.

[121] Reynald LERCIER, "Factoriser des entiers par la méthode des courbes elliptiques", Juin 1993.

[122] S.Duquesnes,"Cryptographie sur les courbes elliptiques", 2005.

[123] Philippe Elbaz-Vincent, "protocole d'échange de clés authentifiés : modèle de sécurité,analyse et construction", 2006.

[124] Ayoub Otmani & Pierre-Louis Cayrel & Damien Vergnaud, "Signatures fondées sur des codes aléatoires".

[125] http ://cayrel.net/IMG/pdf/These_CAYREL_Les_codes_en_cryptographie.pdf

[126] http ://blogperso.univ-rennes1.fr/jeremy.le-borgne/public/Goppa.pdf.

[127] Cameron Mcdonald & Philip Hawkes, and Josef Pieprzyk,"Differential Path for SHA-1 with complexity", 2000.

[128] Diala kheir et Samuel tiberghien,"Etude de l'article :On The Fly Signatures based on Factoring", 2007.

[129] Andreas Enge,"La méthode RSA et la cryptographie fondée sur les courbes elliptiques", 2006.

[130] G.Florin,S.Natkin, "Les techniques de la cryptographie".

[131] Vivien BERNET-ROLLANDE & Simon LALLEMAND, "Etude d'une attaque contre l'algorithme RC4", 2010.

[132] http ://www.tafats.fr/Techniques/Reseaux_de_capteurs/Reseaux_capteurs.html

[133] Susan Landau, "Standing the Test of Time : The Data Encryption Standard",2000.

[134] Ondrej Mikle,"Practical Attacks on Digital Signatures Using MD5 Message Digest",2004.

[135] Olivier Markowitch, "Sécurité des systèmes informatiques : Les signatures digitales".

[136] Z,auctore,"Équations diophantiennes du premier degré",2007.

[137] Stef Graillat,"Un théorème de Brauer",2004.

[138] A.Prodon,"Matrice inverse,Matrices élémentaires",Algèbre Linéaire-Th.M.Liebling,ROSO-EPFL.

[139] pierre alexandre,"distance de Hamminsg",2004.

[140] Robert Cori,,"Algorithmes gloutons".

[141] Xavier Pujol,"Recherche efficace de vecteur court dans un réseau euclidien",2008.

[142] http ://www.memoireonline.com/02/12/5433/m_tat-de-lart-sur-les-reseaux-de-capteurs-sans-fil15.html

[143] Oded Goldreich, "Two Remarks Concerning the Goldwasser-Micali-Rivest Signatur".

[144] Gene Itkis and Leonid Reyzin "Forward-Secure signatures with Optimal Signing and verifying"

Résumé

Les réseaux de capteurs sont des réseaux formés d'un grand nombre de nœuds capteurs qui collaborent entre eux pour fournir un service bien déterminé, dont leurs domaines d'application sont nombreux. Cependant, ce type de réseau ont des ressources limitées ; la limitation des capacités et de traitement, de stockage et ainsi d'énergie.

Néanmoins, beaucoup d'obstacles inhérents à leurs spécificités, parmi ces obstacle, le problème de sécurité se pose avec acuité et doit être solutionner de manière appropriée et en conformité avec les caractéristiques particulières des RCSF, et parmi ces problèmes de sécurité se situe le problème de gestion de clés.

Dans ce mémoire, nous présentons une étude des problèmes de sécurité dans les RCSF ainsi diverses attaques ont été étudiées et pour y faire face, la technique de cryptographie adaptée s'est avérée un bon choix de solution. De plus nous avons ainsi étudier et classifier les différents cryptosystèmes asymétriques proposés, au travers lesquels les buts de sécurité face aux attaques potentielles sont accomplis de manière plus ou moins satisfaisante, de ce fait nous avons élaboré une comparaison entre ces cryptosystèmes suivant des critères bien spécifier dans le but de faire une synthèse d'applicabilité de l'un de ces cryptosystèmes dans les RCSF.

Mots clés **:** Réseaux de capteurs sans-fil, Sécurité, Cryptographie asymétrique.

Abstract

Sensor networks are networks containing a large number of sensor nodes that can work together to provide a service specified property, whose their areas of application are numerous. However, such networks have limited resources, limitations of capacity ,storage and treatment.

Nevertheless, many obstacles inherent to their specific characteristics. Amongst these obstacles, the security problem is an increasing problem who must resolve in accordance with the special characteristics of RCSF, and among these security issues is the problem of key Management.

In this memoir, we present a study of security issues in RCSF and various attacks have been studied and to deal with, the cryptographic technique has proved a good choice of solution. In addition we also study and classify the different asymmetric cryptosystems proposed, through which the goals of security against potential attacks are made more or less satisfactory, thus we developed a comparison between these cryptosystems following criteria clearly specify in order to summarize the applicability of such a cryptosystem in RCSF.

Keywords **:** Networks of wireless sensors, security, Asymmetric cryptography.