République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE

TIZI-OUZOU

FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de Master académique

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Réseaux, Mobilité et Systèmes Embarqués**

Thème :

Vérification automatique de protocoles de sécurité dans les systèmes RFIDs

Vérification automatique de protocoles de sécurité dans les systèmes RFIDs

Mémoire soutenu publiquement devant le jury composé de :

Président :Mr M.Daoui.
Encadreur : Mme Malika BELKADI.
Examinateur : Mme R.Aoudjit.
Examinateur : Mme R.Hadaoui.

Présenté par :

Si-tayeb Fazia

Tahir célia.

Présenté le : 04 /07/ 2018

Remerciement

Nous remercions ALLAH qui nous aide et
nous donne la patience et le courage durant ces longues années d'études.

On tient à remercier chaleureusement
et à exprimer notre profonde reconnaissance et sympathie à nos
professeurs du département d'informatique de Tizi-Ouzou
pour l'élaboration de ce travail.

Nous remercions notre Encadreur
Mme Malika – Belkadi pour son soutien, ses recommandations
judicieuses, sa disponibilité, son suivi attentif,
ses encouragements continus et c'est ce qui nous a permis de mieux
nous exprimer et faire valoir nos connaissances.

Enfin, nous tenons également à remercier
toutes les personnes qui ont participé de près ou de loin
à la réalisation de ce

Célia et Faiza .....

Dédicaces

Je dédie ce modeste mémoire qui est le fruit de nombreuses années d'études et de travail, tout en exprimant ma profonde gratitude et sympathie à toutes les personnes qui ont participé de près ou de loin pour mener à bien ce projet et plus particulièrement :

A ma mère qui me comble d'amour,
A mon père qui m'a toujours épaulé, soutenu financièrement et moralement et donné son amour et son respect.

A tout mes frères , sœurs et ami(e)s
que j'aime beaucoup ;

A mon binôme et mon amie Célia
, celle avec qui j'ai partagé
toutes mes années d'études et à toute sa famille.

Faiza. . . .

Dédicaces

Je dédie ce modeste mémoire qui est le fruit de nombreuses années
d'études et de travail, tout en exprimant
ma profonde gratitude et sympathie à toutes les personnes
qui ont participé de près ou de loin
pour mener à bien ce projet et plus particulièrement :

A ma mère qui me comble d'amour,
A mon père qui m'a toujours épaulé, soutenu financièrement et
moralement et donné son amour et son respect.

A ma sœur, mes frères et ami(e)s
que j'aime beaucoup ;

A mon binôme et mon amie faiza
, celle avec qui j'ai partagé
toutes mes années d'études et à toute sa famille

Célia….

Résumé de mémoire

De nos jours, l'identification et le suivi d'objets se développent de plus en plus, au départ, les codes barres permettaient cette identification mais ne permet pas le stockage d'un grand nombre de données. C'est pourquoi les étiquettes RFID se sont développées ; L'identification par radiofréquence ou RFID (Radio Fréquence Identification) permet d'identifier à distance des objets ou individus, à l'arrêt ou en mouvement, et d'échanger avec eux des données. Conceptuellement, la RFID et le codage à barres sont tout à fait semblables ; tous les deux sont prévus pour fournir l'identification rapide et faible d'articles et des possibilités de filiation. La différence principale entre ces deux technologies est que le codage à barres se lit avec un laser optique et que le lecteur RFID balaye ou interroge une étiquette en utilisant des signaux de fréquence radio.

Les systèmes RFID ne sont pas nouveaux dans leur principe, des dispositifs d'identification d'avions par transpondeur IFF (identification Friend or Foe) ont été utilisés dés la deuxième guerre mondiale

Malgré l'absence d'une véritable standardisation, les étiquettes RFID se développent très rapidement dans des domaines très variés : logistique, identification, contrôle d'accès, protection contre le vole ….etc. Ces développement ont leurs conséquences sur la vie privée ne vont pas sans soulever des problèmes éthiques mais l'avenir des RFID semble quoi qu'il en soit promis au succès.

Une application d'identification automatique RFID se compose d'un lecteur, une ou plusieurs étiquettes et un ordinateur de stockage, le lecteur transmet un signal selon une fréquence déterminée vers une ou plusieurs étiquettes radio situées dans son champ de lecture. Celles-ci transmettent en retour un signal. Lorsque les étiquettes sont « éveillées » par le lecteur, un dialogue s'établit selon un protocole de communications prédéfinies. La description de cette application peut être résumée en référence avec le modèle en couche de l'ISO, en 3 couches principales : couche physique, couche transport et la couche application.

La grande capacité de contenu et la vitesse de marquage sont Les avantages majeurs des étiquettes radio fréquence par rapport au code à barres mais ces avantages ne vont pas sans contraintes : Le cout de conception d'une application RFID restent nettement supérieurs à ceux des étiquettes code a barres. Ainsi que Le manque de standardisation qui présente un vrai inconvénient des systèmes RFID.
Cependant, Il est difficile de parler des systèmes RFID sans parler de la sécurité de l'information et de la protection des données personnelles , La RFID est sujette aux menaces à l'encontre des trois grands domaines de la sécurité : la confidentialité, l'intégrité, et la disponibilité , et comme menaces on peut citer Accès physique au matériel , le clonage et le spoofing ,le Déni de service , L'attaque man-in-the-middle…etc

pour garantir un certain niveau de sécurité dans les systèmes RFID , des différents aspects de sécurité sont utilisés : la cryptologie et l'étude des protocoles cryptographiques tel que le

chiffrement des données (Le cryptage symétrique et asymétrique) , La signature numérique et la fonction de hachage .
Malheureusement, l'usage de méthodes cryptographiques ne suffit pas pour garantir le secret d'une information confidentielle. Tous les jours, de nouvelles failles sont découvertes sur des protocoles cryptographiques, il est donc indispensable de vérifier automatiquement la sécurité des protocoles cryptographiques avant leurs mises en service.

Dans le domaine de vérification automatique des protocoles de sécurité, Il y a plusieurs analyseurs de protocoles, mais la plateforme AVISPA (Automated Validation of Internet Security Protocols and Applications) est l'analyseur le plus connu qui modélise un grand nombre de protocoles.

Dans ce mémoire on a étudié des protocoles d'authentification dans les systèmes RFID qui peuvent être modélisés avec le langage de spécification HLPSL et vérifié automatiquement avec l'outils AVISPA .

Mots-clefs : RFID, Protocoles cryptographiques, Spécification et vérification formelle de protocoles cryptographiques, AVISPA , HLPSL.

Liste des tableaux

Liste des figures

TABLE DES MATIÈRES

Chapitre III : les protocoles de sécurités dans les systèmes RFID

Chapitre IV : la vérification formelle d'un protocole RFID a l'aide d'AVISPA et SPAN

Introduction générale

Parmi les systèmes qui ont été développés rapidement au cours des dernières années, on peut constater ceux d'identification par radiofréquence (RFID), cette technologie permet d'identifier à distance des objets ou individus, à l'arrêt ou en mouvement, et d'échanger avec eux des données.
Jour après jour, l'importance des systèmes d'identification par radiofréquence (RFID) augmente pour ses puissantes capacités identification automatique, localisation et contrôle d'accès des objets. Cependant, les techniques RFID sont en proie à la sécurité et problèmes de confidentialité dus au canal de communication sans fil. C'est pourquoi des moyens cryptographiques sont mis en œuvre afin de garantir des principes essentiels tels que l'authenticité du message, l'intégrité ou encore la disponibilité.

On voit donc se créer de nombreux protocoles cryptographiques permettant de sécuriser les données envoyées , Mais sur ces protocoles, les chercheurs ont découvert des failles de sécurité, il est donc important de vérifier automatiquement la sécurité des protocoles cryptographiques avant leurs mises en service car la sécurité de ces protocoles n'est pas garantie par l'usage des méthodes de chiffrement seulement, mais aussi par une vérification automatique et formelle , cette vérification se fait a l'aide des outils de vérification automatique qui s'appuient sur le modèle formelle tel que AVISAP , proVerif ,Casper et Hermes.

Le sujet de notre projet est la vérification formelle et automatique d'un protocole de sécurité dans les systèmes RFID à l'aide de l'outil AVISPA , notre travail est structuré comme suit :

Chapitre I : Généralités sur les systèmes RFID

Dans ce chapitre nous avons vu d'abord la technologie code-barres ensuite nous avons présenté la technologie RFID : son évolution, ses composants, et son principe de fonctionnement, ses différents avantages et inconvénients et ses champs d'application.
A la fin du chapitre nous avons comparé les deux technologies : technologie RFID et la technologie code-barres.

Chapitre II : La sécurité dans les systèmes RFID

Dans ce chapitre nous avons défini les services de sécurité, les mécanismes de sécurité , leurs principes de fonctionnement et leurs utilisations, ensuite nous avons vu quelques attaques possibles qui peuvent affecter les systèmes RFID et quelques contre mesure .

Chapitre III : les protocoles de sécurités dans les systèmes RFID

Dans ce chapitre, nous avons vu quelque protocole d'authentification dans les systèmes RFID, Ensuite, nous avons proposé un protocole d'authentification qui utilise des générateurs de nombres pseudo-aléatoires (PRNG) et quelques opérations cryptographiques simples.

Chapitre IV : la vérification formelle d'un protocole RFID a l'aide d'AVISPA et SPAN

Dans ce chapitre Nous présentons la vérification formelle, ces caractéristiques et ces méthodes et les outils de vérification automatique qui s'appuient sur ce modèle de vérification. Nous concentrons sur l'outils AVISPA .
Ensuite on verra le langage formel afin de vérifier la sureté de protocole propose dans le chapitre 3 a l'aide de d'AVISPA.

Chapitre I :

Généralités sur les systèmes RFID.

I.1.Introduction :

Parmi les systèmes d'identification automatique les plus utilisés dans notre vie quotidienne : les systèmes RFID (Radio Frequency Identification) qui utilise le rayonnement radiofréquence pour la traçabilité des objets porteurs d'étiquettes lorsqu'ils passent à proximité d'un interrogateur.

Dans ce chapitre nous allons voire d'abord un autre type de systèmes d'identification automatique qui est le code-barres ensuite nous allons entamer la technologie RFID, donc nous allons donner son évolution, une description détaillée de ses composants, et son principe de fonctionnement, ensuite nous présenterons les différents avantages et inconvénients de cette technologie ainsi que ses champs d'application.
A la fin du chapitre nous allons comparer les deux technologies : technologie RFID et la technologie code-barres.

I.2. le code-barres :

Le code-barres est l'un des **systèmes d'identification automatique** qui est la représentation graphique d'une donnée numérique (nombres) ou d'une donnée alphanumérique (lettres et nombres) sous forme d'un symbole constitué de barres plus ou moins épaisses et d'espaces.[1][2]

I.2.1 Les différentes familles de codes-barres :

On distingue principalement deux familles de codes-barres :

Les codes-barres à une dimension (1D)

Figure I.1 : Les codes-barres

Figure I.2 : Les codes-barres UPC

Les codes-barres à deux dimensions (2D)

Figure I.3 : Les codes QR

Figure I.4 : Les codes-barres PDF417

I.2.3 La différence entre le code-barres et les systèmes RFID :

- Les lecteurs de codes-barres ne peuvent traiter qu'une seule étiquette à la fois, tandis que dans les RFID , le lecteur peut recevoir plusieurs tags en une seule seconde.
- Les lecteurs codes-barres utilisent un capteur et une lumière pour lire les données sur l'étiquette tandis que la RFID utilise des ondes radio qui n'ont pas besoin de ligne directe.
- Les code-barres sont vraiment simples et peuvent être facilement reproduits ou contrefaits, tandis que la RFID est plus complexe.
- Les tags RFID peuvent être cachés pour se protéger de l'environnement alors que les code-barres doivent être exposés.
- À la différence du code-barres qui doit être vu et absolument positionné face à un lecteur pour pouvoir être lu, la RFID doit seulement se trouver dans le champ électromagnétique d'un lecteur, ce qui signifie qu'elle peut être lue qu'elle que soit sa position, son orientation et son environnement.
- L'inconvénient majeur de la RFID est son prix parce que les code-barres n'utilisent que du papier, il est nettement moins cher que les étiquettes RFID qui sont des petits circuits intégrés.
- En cas de code-barres la distance entre le support de données et le lecteur est entre 0 – et 50 cm tandis que la RFID cette distance est entre 0 et 5 m

I.3 Historique de la RFID :

L'évolution de la RFID est passée par plusieurs phases : [1][3][6]

1940

Le système IFF pour "Identify : Friend or Foe", est la première utilisation de la RFID.*le IFF* est utilisé pour la première fois lors de la Seconde Guerre Mondiale pour identifier si les avions qui arrivaient dans l'espace aérien britannique étaient amis ou ennemis les alliés mettaient en place dans leurs avions des transpondeurs afin de répondre aux interrogations de leurs radars.

1970

Durant les années 1960-1970, les systèmes RFID restent une technologie confidentielle, à usage militaire pour le contrôle d'accès aux sites sensibles, notamment dans le nucléaire.

1980

Les avancées technologiques permettent l'apparition du tag passif. Le tag RFID rétromodule l'onde rayonnée par l'interrogateur pour transmettre des informations. Cette technologie permet de s'affranchir de source d'énergie embarquée sur l'étiquette réduisant de ce fait son coût et sa maintenance.

1990

Début de la normalisation pour une interopérabilité des équipements RFID.

1999
Fondation par le MIT (Massachusetts Institute of Technology) de l' Auto-ID center : centre de recherches spécialisées en identification automatique .
2004
L'auto-ID du MIT devient "EPCglobal", une organisation chargée de promouvoir la norme EPC (Electronic Product Code), extension du code barre à la RFID.

A partir de 2005
Les technologies RFID sont largement répandues dans quasiment tous les secteurs industriels (aéronautique, automobile, logistique, transport, santé, vie etc.). L'ISO (International Standard Organisation) a largement contribué à la mise en place de normes tant techniques qu'applicatives permettant d'avoir un haut degré d'interopérabilité voire d'interchangeabilité.

2009
Création du Centre National de Référence RFID.

I.4 Les composants des systèmes RFID :

Dans tout système RFID, on retrouve les mêmes composants de base (figure1) : un lecteur, une ou plusieurs étiquettes et un ordinateur de stockage et de traitement des informatisions recueillies par le lecteur [7] [8].

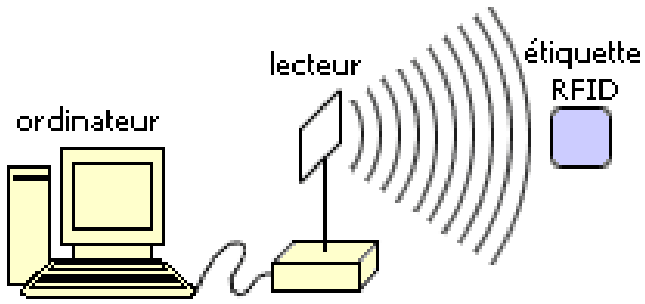


Figure I.5 : Schéma générale d'un système d'identification automatique(RFID)

I.4.1 Lecteurs RFID :

Ce sont des dispositifs actifs, émetteurs de radiofréquences qui vont activer les tags qui passent devant eux en leur fournissant l'énergie dont ils ont besoin pour fonctionner.
L'interrogateur (lecteur) envoie des commandes particulières auxquelles répond , le tag des données peuvent ainsi être échangées. [14]

Figure I.6 : exemples de lecteurs RFID.

Les lecteurs RFID peuvent être fixe ou portable selon l'utilisation à laquelle il sera destiné :

- **Le lecteur RFID fixe :** comme son nom l'indique il est installé de manière fixe et ne peut donc pas être transporté pour la lecture des puces à distance
- **Le lecteur RFID portable :** dans le cas d'un lecteur portable, les objets n'ont plus besoin d'être transportés à proximité du lecteur, c'est le lecteur qui se déplace.

Le lecteur le plus utilisé est le lecteur fixe, mais il peut également prendre la forme d'un lecteur portable.

I.4.2 l'étiquette (Tag) RFID :

C'est un dispositif récepteur, que l'on place sur les éléments à identifier (objet, animal...). Ils sont munis d'une **puce** contenant les informations et d'une **antenne** pour permettre des échanges d'informations. [6] [12] [13]

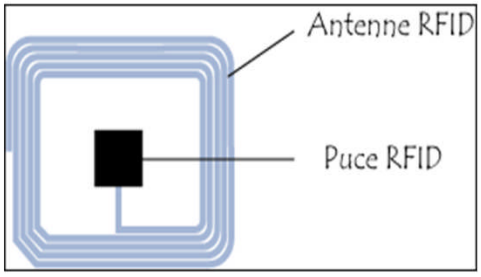


Figure I.7: composants d'une étiquette RFID

On distingue trois types de tags RFID :

Les tags passifs : sont les tags RFID les plus économiques et les plus généralement utilisés dans les applications RFID.
les tags de ce type ne sont pas équipés de pile interne, ils obtiennent leur énergie des lecteurs RFID. Le lecteur RFID envoie des ondes électromagnétiques à l'antenne du tag, qui va réagir (se réveiller) et renvoyer un signal au lecteur en utilisant l'énergie de ces ondes reçues.

Les étiquettes passives les plus utilisées actuellement sont les EPC (Code Produit Electronique)

Les tags actifs : utilisent leur propre énergie pour émettre leurs ondes, en utilisant une pile interne. Ils peuvent ainsi avoir une très longue distance de lecture. Ils sont plus onéreux que les tags passifs et sont donc généralement utilisés pour tracer des articles de valeur. Ils ont de meilleures portées, de meilleures capacités de calcul et des mémoires plus importantes, mais ils ont aussi une espérance de vie plus courte et plus chers à produire.

Les tags semi-passifs : petits et légers, sont des tags intermédiaires entre les tags actifs et les tags passifs. Ils utilisent généralement une pile comme source d'énergie pour son alimentation interne (comme les tags actifs), et pour transférer des données elles utilisent l'énergie générée par les ondes des lecteurs RFID (comme les tags passifs)

La puce :

Il existe trois modes de fonctionnement de tags RFID selon la programmation de la puce RFID :

Puce en **«lecture seule»** : Elle comporte un numéro d'identification gravé par le fondeur dès la fabrication de la puce, Le numéro peut être lu mais il n'est plus modifiable.

Puce en **«écriture une fois, lecture multiple»**. Le tag est livré vierge L'utilisateur peut enregistrer son numéro d'identification unique lors de la première utilisation de l'étiquette. Ensuite, il est seulement possible de lire cette information.

Puce en **«lecture réécriture»**. Dans ce mode de fonctionnement, le tag peut être livré vierge ou avec des informations, mais les informations peuvent être effacées et réécrites par l'acheteur du tag presque autant de fois qu'il le souhaite.

L'antenne RFID :

L'**antenne RFID** est un élément primordial du système RFID qui est généralement intégrée au lecteur et à l'étiquette RFID . Elle permet d'activer les tags afin de recevoir des données et d'en transmettre les informations. [5]

La plupart des lecteurs intègrent une antenne RFID, mais pour les lecteurs de moyenne et de longue portée l'ajout d'une ou de plusieurs antennes RFID externes s'avère nécessaire afin de capter les informations contenues dans la puce RFID à travers de longues distances.

Deux types principaux d'antennes se distinguent :

- **les antennes intégrées :** elles sont intégrées au lecteur, leur utilisation est conseillée pour les lecteurs de basse fréquence à portée limitée ;
- **les antennes externes** : elles ne font pas partie du lecteur, elles sont plus puissantes et s'avèrent donc utiles pour avoir une plus grande portée.

Fréquences :

Fonctionnant avec les ondes radio, la **RFID** utilise plusieurs types de fréquence : [16]

Basses Fréquences (LF): 125 KHZ , cette gamme de fréquence est utilisée par les tags passifs avec une distance de lecture de quelques centimètres ;

Hautes Fréquences (HF) : de 13,56 MHz , utilisée par les tags passifs pour une distance de lecture de un à plusieurs mètres.

ultra Hautes Fréquences (UHF): de 800 – 930 MHz , cette gamme de fréquence est utlisée par les tags passifs , actif et semi actifs.

Super hautes Fréquences (SHF) : 2,45 GHZ / 5,8 GHZ , utilisée par les tags passifs et actifs [3]

Les différents supports de tag RFID :

Cartes et badges | **Bracelets** | **Etiquettes et stickers** | **Porte-clés et tags RFID** | **Puces sous cutanés**

Figure I.8: Exemples de support RFID

I.4.3 Le couplage tag /lecteur RFID :

La liaison entre le tag et l'interrogateur se réalise par :

- **Couplage magnétique :** dans le cas d'un champ proche (quelques cm à 1,5 m). L'interrogateur utilise alors des BF (Basses Fréquences) ou des HF (Hautes Fréquences).
- **Couplage électrique :** dans le cas d'un champ lointain (jusqu'à 6m). L'interrogateur utilise alors des UHF (Ultra Hautes Fréquences) ou des SHF (Super Hautes Fréquences).

I.4.4 Principe de fonctionnement des systèmes RFID :

- Le lecteur envoie une onde électromagnétique porteuse d'un signal en direction des objets à identifier et demande des informations.
- L'étiquette fixée sur ces objets réagit à la réception du signal envoyé par le lecteur en renvoyant vers ce dernier les informations demandées.
- Le lecteur envoie les informations à un ordinateur de stockage et de traitement (serveur).

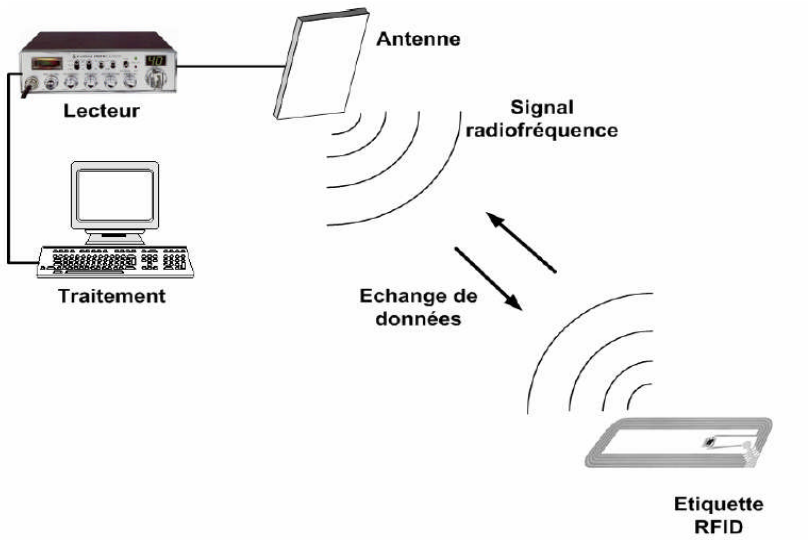


Figure I.9: Principe de fonctionnement d'un système RFID.

I.5. Les avantages et inconvénient de la RFID :

Les avantages et les inconvénients les plus essentiels des systèmes RFID sont :

I.5.1 Les avantages de la RFID :

Si la RFID est en passe de devenir la technologie de traçabilité logistique la plus utilisée, c'est parce qu'elle offre de nombreux avantages.

Au nombre de ses principaux avantages, on a :

- Efficace contre le vol
- Lecture sans contact ni visée avec la marchandise
- Nombre plus important de données stockées par rapport aux code-barres.
- Possibilité de lire plusieurs étiquettes simultanément
- Évite les erreurs de saisie et fournit des données fiables

I.5.2 Les inconvénients de la RFID :

- **Le manque de standardisation :** par exemple, les États-Unis et l'Europe ont une gamme de fréquences différente à laquelle les étiquettes RFID fonctionnent Cela oblige les organisations à être conscientes du mode de travail dans les autres pays.

- **Les coûts élevés de mise en œuvre:** Le coût est le plus grand obstacle dans l'utilisation des étiquettes RFID pour le suivi des produits à faible coût. Les systèmes RFID sont généralement plus chers que les systèmes alternatifs tels que les systèmes de codes à barres.
- **La collision de tag et de lecteur :** La collision des étiquettes se produit lorsque de nombreuses étiquettes sont présentes dans une zone confinée. Le lecteur d'étiquettes RFID alimente plusieurs étiquettes simultanément, toutes reflétant leurs signaux vers le lecteur. Il en résulte une collision entre étiquettes et le lecteur RFID ne parvient pas à différencier les données entrantes.

La collision du lecteur RFID se produit lorsque la zone de couverture gérée par un lecteur RFID chevauche la zone de couverture d'un autre lecteur. Cela provoque une interférence de signal .

- Il est difficile pour un lecteur RFID de lire les informations dans le cas où les étiquettes sont installées dans des produits liquides ou métalliques. Le problème est que les surfaces liquides et métalliques ont tendance à refléter les ondes radio, ce qui rend les étiquettes illisibles

- **Le respect de la vie privée :** Les consommateurs ont tendance à s'inquiéter de leur vie privée lorsqu'ils achètent des produits avec ces étiquettes, car on croit qu'une fois que les puces radios sont installées dans un produit, il continue de suivre une personne, et ses renseignements personnels peuvent être recueillis et transmis au lecteur. Ainsi, alors que de nombreux magasins affirment qu'ils désactivent les étiquettes après l'achat du produit, les acheteurs continuent d'être inquiets de cette technologie.

- **L'impact des ondes-radio sur la santé :** les puces RFID attachées ou implantées dans le corps humain de manière sous cutanée en vue d'assurer leur suivi médical présentent un grand risque sur la santé des patients.

Quoiqu'il en soit, la **RFID** demeure à ce jour l'une des meilleures solutions de traçabilité. Le fort engouement qu'elle suscite devrait permettre à terme de faire baisser le prix unitaire des étiquettes, rendant ainsi la technologie plus accessible.

I.6. Exemples d'applications des systemes RFID:

Tous les jours nous utilisons des produits RFID sans le savoir : à travers des cartes de transports, des étiquettes antivol dans les magasins, les clés des véhicules récents, des badges de sécurité, les puces de marquages des animaux domestiques, les cartes bancaires de paiement sans contact. Tous ces objets du quotidien fonctionnent grâce à une technologie RFID qui a pour avantage **de faire gagner du temps aux usagers** et de permettre une **lecture rapide des données** [19] [20]

I.6.1 Le suivi des animaux :

Des applications de plus en plus nombreuses de traçabilité des animaux se développent, que ce soit les étiquettes auriculaires sur les animaux d'élevage ou les étiquettes sous cutanées pour les animaux domestiques. Dans tous les cas, il s'agit d'accroché a l'animal un tag RFID qui permet de constituer des historiques des différentes activités, de son alimentation, ou encore de son état de santé.

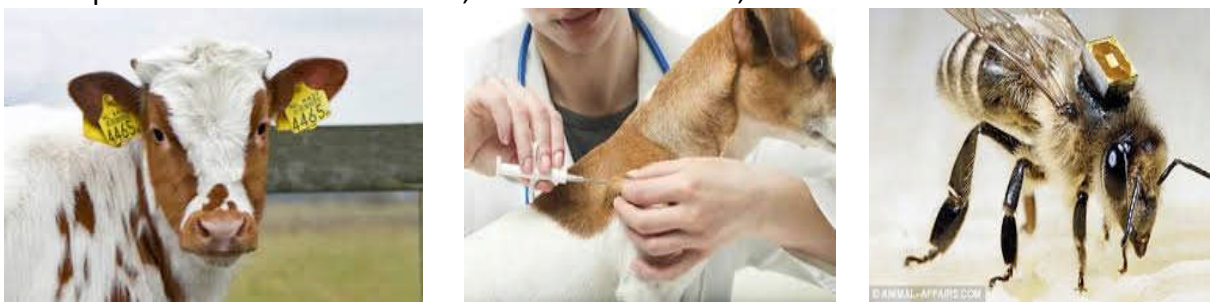


Figure I.10: RFID et le suivi des animaux.

I.6.2 Le contrôle d'accès :

Les étiquettes RFID sont utilisées pour le contrôle d'accès des immeubles ou des parkings : Un lecteur ouvre l'accès au local, automatiquement ou par la personne qui le contrôle, un ordinateur permet de gérer et d'enregistrer les coordonnées des personnes et/ou les véhicules qui se présentent ,Ces enregistrements peuvent être exploités ultérieurement.

Figure I.11: RFID et le contrôle d'accès.

I.6.3 Le suivi et le tri des bagages :

Capables d'être lues à distance, les étiquettes RFID permettent à la compagnie aérienne d'assurer un suivi en temps réel du bagage, limitant ainsi le nombre de bagages qui ratent leur vol. Elles détiennent en

effet une capacité d'identification nettement supérieure à celle des codes-barres et autres étiquettes papier actuellement utilisés dans les aéroports.

Figure I.12: RFID dans le suivi des bagages.

I.6.4 La RFID dans le secteur de la santé :

L'adoption de systèmes de suivi RFID a été utile dans la réduction des efforts et des erreurs humaines dans la maintenance et le contrôle de la gestion d'inventaire, mais aussi dans l'amélioration de la sécurité du patient et la qualité des prestations.

I.13: RFID dans le secteur de la sante.

Les applications de la RFID dans le secteur de la santé couvrent de nombreux domaines, allant de la chaine d'approvisionnement, notamment visant les matériels à haute valeur comme les pacemakers qui nécessitent un très haut niveau de traçabilité. La sécurité des patients nécessitent par ailleurs une identification assurée par des tags ou bracelets RFID . D'autres applications visent à améliorer le parcours du patient dans un établissement hospitalier ou encore faciliter l'accès du personnel soignant aux informations médicales. C'est dire à quel point la RFID joue et jouera un rôle croissant et déterminant dans la gestion de la santé en améliorant singulièrement l'efficacité mais aussi en réduisant les coûts de gestion.

I.6.5 La RFID dans les documents d'identité :

Le passeport ainsi que la CNI (Carte National d'Identité) sont maintenant équipées de puce RFID (carte à puce sans contact). Cette puce RFID contient un certain nombre d'informations relative a son

porteur dont la photo et les empreintes digitales ainsi que des certificats numériques. Elle sert non seulement à identifier le porteur (empreinte digitale et reconnaissance faciale) mais elle constitue également l'un des éléments de sécurité permettant de vérifier l'authenticité du document

Figure I.14: la RFID dans les documents d'identité

I.7. Les normes RFID :

Le développement des normes pour les systèmes RFID est nécessaire pour la maitrise de cette technologie et le développement de son marché. Le but de la normalisation vise à assurer le fonctionnement, l'interopérabilité et l'interchangeabilité des systèmes RFID.
Deux normes sont disponibles pour les utilisateurs [17]:

La normes d'ISO :

La normalisation des protocoles de communication entre tags et lecteurs est le rôle du comité technique de l'ISO (International Organisation for Standardisation) qui a rédigé un certain nombre de normes qui réglemente les paramètres de communication, comme la fréquence de fonctionnement, la bande passante, la puissance d'émission maximale, le débit, le protocole de communication. L'ISO a rédigé aussi des normes relatives à l'identification et la gestion des objets ou équipements dans la série des protocoles d'interface ISO 18000 conçus pour des opérations de logistique. Ces normes couvrent toute la gamme des fréquences utilisées dans le monde en matière de RFID. Les sept éléments de cette norme sont décrits dans le tableau suivant[16].

Références	Fréquences concernées	Intitulé	Edition
18000-1		RFID pour la gestion d'objets-partie1 : Architecture de référence et définition des paramètres à normaliser .	13/09/2004
18000-2	<135 KHz	RFID pour la gestion d'objets-partie2 : Paramètres de communications d'une interface d'air en dessous de 135kHz.	13/09/2004
18000-3	13,56 MHz	RFID pour la gestion d'objets-partie3 : Paramètres de communications d'une l'interface d'air à 13,56 MHz.	13/09/2004
18000-4	2.45 GHz	RFID pour la gestion d'objetspartie4 : paramètres de communications d'une l'interface d'air à 2,45 GHz	31/08/2004
18000-5	5.8 GHz	RFID pour la gestion d'objets-partie5 : Paramètres de communication d'une interface d'air pour les systèmes RFID exploités à 5.8 GHz. Le but de cette norme était de définir la couche physique, le système anti collision et les valeurs des protocoles RFID exploités dans la bande des 5.8 à 5.9 GHz. Ce thème de normalisation a été abandonné.	Abandonnée
18000-6	960 MHz	RFID pour la gestion d'objets-partie6 : Paramètres de communications d'une interface radio entre 860 MHz et 960 MHz.	31/08/2004
18000-7	433 MHz	RFID pour la gestion d'objets-partie7 : Paramètres de communications de l'interface radio pour les systèmes RFID passifs exploités à 433 MHz.	15/01/2008

Tableau I.1 : Norme ISO 18000 pour la standardisation des systèmes RFID**.**

Le standard EPC

Le **Code Produit Electronique**, ou ***EPC*** pour *Electronic product code*, est un identifiant unique permettant d'identifier un objet dans une chaîne de production il est parmi les applications des RFID les

plus longtemps utilisées ; Il s'agit d'une tentative de créer un réseau global, normalisé, permettant d'étiqueter et de suivre tout ce qui peut être
expédié, stocké ou vendu . Cette norme a été élaborée par l'Auto-ID Center (un centre de recherche) et des laboratoires dans les plus prestigieuses universités du monde.

l'EPC est un peu plus long que le code à barres, au lieu d'être imprimé sous la forme de barres parallèles, il est stocké dans un tag RFID, est un numéro unique attribuable à chaque objet, ce code peut servir à l'obtention d'informations (statuts, localisation,…) via le réseau EPC network .

La structure du code EPC 96bits :

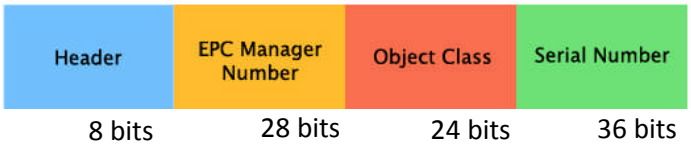


Figure I.15: la structure du code EPC 96bits

Header : version du standard EPC utilisé.
EPC Manager Number : le code du fabriquant.
Object Class : identifier le type de produit.
Serial Number : numéro unique qui représente les informations relatives au produit (numéro de série)

I.8. Conclusion :

Dans ce chapitre, nous avons présenté les systèmes RFID : leurs composants, leur principe de fonctionnement, quelques normes et standard, Les gammes de fréquence, et quelques domaines d'application ainsi que les avantages et inconvénients de la RFID.

L'identification par radio fréquence est de plus en plus utilisé mais malgré ses puissances, la Sécurité reste le principal frein à une large adoption de cette technologie, plusieurs contraintes rendent la sécurisation dans ces systèmes complexe.

Dans le chapitre suivant, nous allons voir les problèmes de sécurité dans les systèmes RFID et les contre-mesures mises en place pour faire face aux attaques sur les systèmes RFID.

Chapitre II :

La sécurité dans les systèmes RFID

II.1 Introduction :

Le système d'informatique représente un patrimoine essentiel de l'organisation, qu'il convient de protéger. La sécurité informatique consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu ,et pour bien comprendre cet aspect nous allons présenter dans ce chapitre les différents services de sécurité,puis on donnera une descriptiondétaillée sur les mécanismes de sécuritétel que le chiffrement des données et la signatures numériques, leurs principes de fonctionnement et leurs utilisations, ensuitenous allons essayer de découvrir certaines attaques possibles qui peuvent affecter ces systèmes RFID en considérant le point d'attaque, leur classification et nous allons discuterdes mesures de sécurité possibles qui peuvent être utilisés pour lutter contre ces attaques.

II.2 Sécurité des systèmes informatiques :

Pour qu'un système soit sécurisé, les propriétés suivantes doivent être assurées [22] [23] :

La disponibilité : le système doit fonctionner sans faille durant les plages d'utilisation prévues et garantir l'accès aux services et ressources installées avec le temps de réponse attendu.

L'intégrité : les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets.

Confidentialité : seules les personnes autorisées peuvent avoir accès aux informations qui leur sont destinées. Tout accès indésirable doit être empêché.

D'autres aspects peuvent aussi être considérés comme objectifs de sécurité des systèmes d'information, tels que :

L'authentification: l'identification des utilisateurs est fondamentale pour gérer les accès aux espaces de travail pertinents et maintenir la confiance dans les relations d'échange

Le contrôle d'accès :dans le contexte de la sécurité, le contrôle d'accès est la faculté délimiter et de contrôler l'accès à des systèmes et des applications via des maillons de communication. Pour accomplir ce contrôle, chaque entité essayant d'obtenir un accès doit d'abord être authentifiée, de telle sorte que les droits d'accès puissent être adaptés à son cas.

La non-répudiation: aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

II.3. Mécanismes de sécurité :

La sécurité est un ensemble de stratégies, conçues et mises en place pour détecter, prévenir et lutter contre une attaque. Actuellement, il existe beaucoup de mécanismes de sécurité [26] [27][28].

II.3.1 cryptage ou le chiffrement des données :

Le mot «Cryptographie» est composé des mots grecques : CRYPTO = caché GRAPHY = écrire. C'est donc l'art de l'écriture secrète et une science permettant de préserver la confidentialité des échanges.

Autrement dit c'est un mécanisme de sécurité, qui consiste à traduire un message clair, dit original en un message incompréhensible, inintelligible, le résultat du processus de cryptage est appelé «texte chiffré ou message codé», le processus de cryptage reposent à la fois sur des algorithmes puissants et sur des paramètres appelés clés, c'est ainsi que les techniques de cryptographie sont essentiellement scindées en deux :

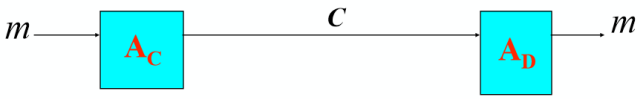


Figure II.1: schéma de la cryptographie

A_C : Algorithme de chiffrement

A_D : Algorithme de déchiffrement

M : message clair

C : texte chiffré

II.3.1.1 Le cryptage symétrique : il consiste à utiliser la même clé pour crypter et décrypter un message. Il est important de savoir que le cryptage symétrique est moins sécurisé, du fait que c'est la seule clé qui est échangée entre les deux entités de communicantes. D'où l'interception de la clé lors de l'échange peut compromettre la sécurité du message crypté.

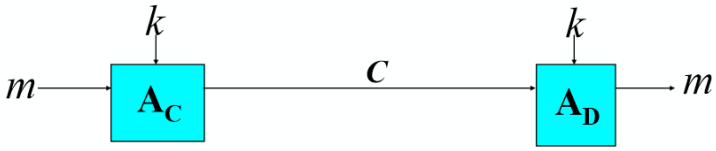


Figure II.2: schéma de la cryptographie symétrique.

Le chiffrement symétrique se déroule en 3 étapes, de la manière suivante :

1ère étape :

- ✓ Génération de la clé secrète par l'émetteur.
- ✓ Envoi de cette clé secrète au récepteur, de manière sécurisée.

2éme étape :

- ✓ Chiffrement du message par l'émetteur, avec la clé secrète.
- ✓ Envoi de ce message chiffré au récepteur.

3éme étape :

- ✓ Réception du message chiffrée par le récepteur.
- ✓ Déchiffrement du message avec la clé secrète reçue auparavant

II.3.1.2 Le cryptage asymétrique : Dans ce type de cryptage les clés existent par paire, c'est-à-dire une clé publique pour le cryptage, et une clé dite secrète pour décrypter le message, et seul l'utilisateur à qui le message est destiné possède la clé secrète pour décrypter le message (l'expéditeur utilise la clé publique du destinataire pour coder son message. Le destinataire utilise sa clé privée pour décoder le message de l'expéditeur) ce mécanisme est plus sécurisé que le cryptage symétrique.

Parmi les chiffrements asymétriques bien connus, on trouve l'algorithme de Diffie-Hellman et RSA .

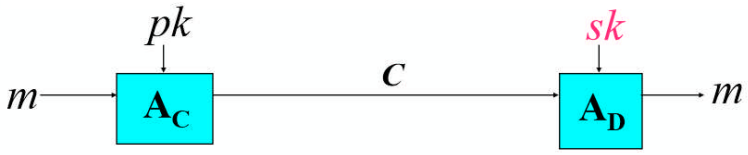


Figure II.3: schéma de la cryptographie asymétrique.

Objectifs :

Parmi les objectifs de la cryptographie :

- Garantir la confidentialité
- Vérifier l'intégrité des données

II.3.2 La fonction de hachage :

Une fonction de hachage (parfois appelée fonction de condensation) est une fonction permettant d'obtenir un condensé (appelé aussi condensat ou haché ou en anglais message digest) d'un texte, c'est-à-dire une suite de caractères assez courte représentant le texte qu'il condense.

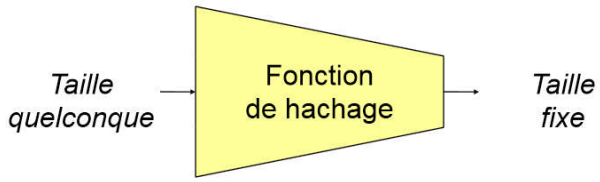


Figure II.6: la fonction de hachage.

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). D'autre part, il doit s'agir d'une fonction à sens unique (one-way function) afin qu'il soit impossible de retrouver le message original à partir du condensé.

Son utilisation :

- **Vérification de l'Intégrité des fichiers ou des messages :** la modification d'un fichier lors d'une transmission peut être prouvée en comparant la valeur de hachage du fichier avant et âpres la transmission.

- **Vérification de mots de passe :**une façon de réduire la violation de mot de passe stocké est de stocker seulement la valeur de hachage de celui-ci.

Pour authentifier un utilisateur, le mot de passe fourni par ce dernier est haché et Comparé avec la valeur de hachage stocké.

- **Signature numérique.**

II.3.3La signature numérique :

Dite aussi signature électronique est un mécanisme permettant de garantir **l'intégrité** d'un document électronique une fois signé et d'en authentifier le signataire, par analogie avec la signature manuscrite d'un document papier,la signature numérique n'a pas d'apparence visuelle comme une signature manuscrite.

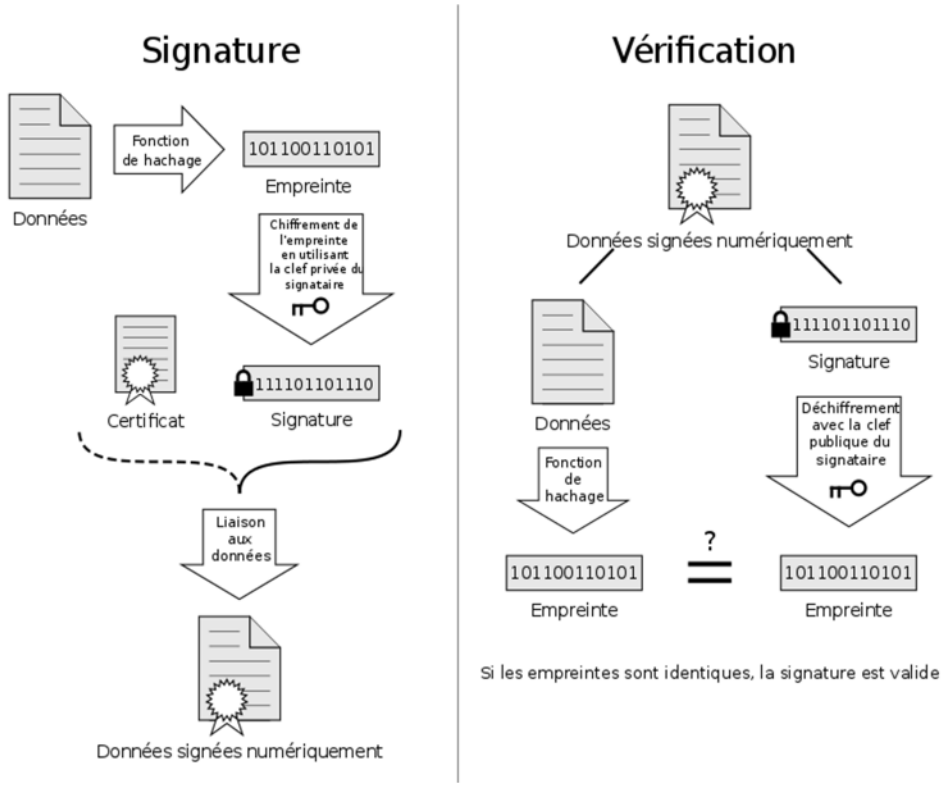
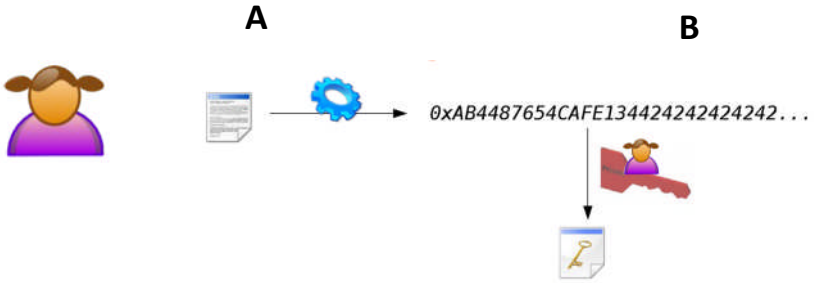


Figure II.5: la signature numérique.

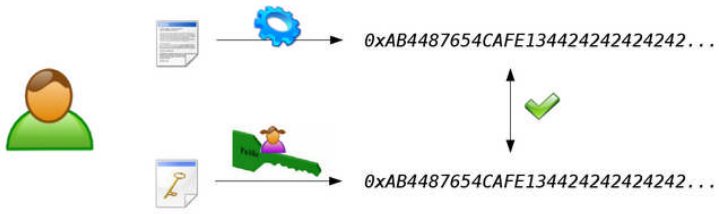
- ✓ Tout d'abord, l'émetteur(A) génère l'empreinte du document au moyen d'une fonction de hachage.
- ✓ Puis, il crypte cette empreinte avec sa clé privée.



- ✓ L'émetteur(A) obtient ainsi la signature de son document.il envoie donc ces deux éléments au récepteur (B)

- ✓ Pour vérifier la validité du document, le récepteur(B) doit tout d'abord déchiffrer la signature en utilisant la clé publique de l'émetteur.
- ✓ Ensuite, (B) génère l'empreinte du document qu'il a reçu, en utilisant la même fonction de hachage que (A)

✓ Puis, il compare l'empreinte générée et celle issue de la signature



- Si les deux empreintes sont identiques, la signature est validée. Nous sommes donc sûr que :

  - C'est l'émetteur(A) qui a envoyé le document,
  - Le document n'a pas été modifié depuis que (A) l'a signé.

- Dans le cas contraire, cela peut signifier que :

  - Le document a été modifié depuis sa signature par (A),
  - Ce n'est pas ce document que (A) a signé.

Une **signature électronique** permet de :

- Authentifier le signataire
- Garantir l'intégrité du document
- Assurer la non-répudiation, c'est-à-dire que l'émetteur du document ne peut pas nier l'avoir envoyé.

II.4.Les attaques informatiques :

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables.

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

Les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.

Les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'il l n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables.

II.5. Classification des attaques RFID :

Avant de proposer et de mettre en œuvre des mesures de sécurité, il est important de Comprendre et de classer les attaques et les menaces existantes dans un système. Dans les systèmes RFID plusieurs auteurs ont classé les attaques selon plusieurs critères citons:

Avoine et al [31] ont intéressé seulement aux menaces sur la vie privée, ils ont classé ces menaces en deux catégories: les fuites d'informations et la traçabilité. Ils ont montré la relation entre la traçabilité et les trois couches de communication RFID (couche physique, couche transport et couche application).

Garfinkel et al[32] ont aussi mis l'accent sur la vie privée et divisé les menaces en deux catégories: les menaces pour la sécurité des données d'entreprise et les menaces personnel.

Karygiamis et al[33] ont proposé un modèle de classification détaillé des risques RFID. Ils ont divisé les risques en trois catégories: les risques basés sur les réseaux, les risques commerciaux et les risques d'intelligence d'affaires.

Dautres auteurs[34] ont proposé une classification des menaces de sécurité RFID qui a deux niveaux. Le premier niveau est de classer les menaces selon les trois couches dans le modèle de communication: menaces de la couche application, de la couche de transport et de la couche physique. Tandis que dans le deuxième niveau, les menaces sont les attaques spécifiques au système.

Mirowski et al[35] ont proposé une classification selon le comportement d'attaquant, mais cette classification est destinée uniquement pour la couche matériel RFID et des séquences du modèle attaquant.

Mitrokosta et al[36] ont divisé les menaces RFID en quatre couches principales: la couche physique, La couche transport, la couche application et la couche stratégique. En outre, il sont créé une catégorie distincte d'attaques multicouches qui exploitent les vulnérabilités à partir de plusieurs couches.

Mitrokosta et al[37]ont spécifié trois catégories principales des attaques RFID basées sur la partie du système qu'ils visent (les attaques affectes à la couche matériel RFID, à couche de communication et$$$ à la couche back-end), et ont subdivisé chaque couche en trois
groupes principaux en fonction des propriétés de la sécurité (la confidentialité, l'intégrité et la disponibilité).

Tandis que d'autres auteurs ont classé les menaces en trois catégories en fonction des objets et des moyens. La première catégorie est les menaces physiques dans lesquelles une menace utilise des moyens physiques pour attaquer le système. La deuxième catégorie est les menaces du canal, ce genre d'attaques est réalisé sur des canaux non sécurisés entre les étiquettes et les lecteurs. La troisième catégorie est des menaces systèmes, comprend les attaques qui affectent le protocole d'authentification et l'algorithme de chiffrement.

II.6. Attaques et contre-mesures dans les systèmes RFID :

Malheureusement, les entreprises et les gouvernements ne sont pas les seuls à s'intéresser à la RFID, les groupes de défense des libertés civiles, les hackers et les criminels s'intéressent aussi vivement à ce nouveau développement, bien que pour des raisons très différentes.
La RFID est sujette aux menaces à l'encontre des trois grands domaines de la sécurité :
la confidentialité, l'intégrité et la disponibilité. Les attaques peuvent être le fruit de ces menaces plus ou moins sophistiquées, et utilisées à la fois contre plusieurs couches du système.

Dans cette partie on va utiliser la classification des auteurs qui ont proposé une classification des menaces de sécurité RFID qui a deux niveaux.

- Le premier niveau est de classer les menaces selon les trois couches dans le modèle de communication: menaces de la couche application, de la couche de transport et de la couche physique.
- Tandis que dans le deuxième niveau, les menaces sont les attaques spécifiques au système.

II.6.1 Attaques de la couche matérielle :

La couche matérielle dans les communications RFID est constituée de l'interface physique, des signaux radios utilisés, et des matériels RFID. Les adversaires de cette couche profitent de la nature sans fil des communications RFID, de leur faible protection physique et leurs lacunes en termes de résistance contre les manipulations physiques. Voici quelques attaques de cette couche[33] [38] [39] :

Accès physique au matériel :

Les systèmes RFID fonctionnent seulement quand les étiquettes RFID et les bases de données internes sont disponibles. Une opération sur une étiquette peut être interrompue en bloquant intentionnellement son accès et ainsi priver le lecteur des données présentes sur l'étiquette. Il existe plusieurs techniques plus ou moins élaborées qui permettent de provoquer ce cas de figure : Retirer ou cacher l'étiquette RFID d'un objet empêche toute détection de ce dernier. Cette attaque peut être utilisée pour dérober des articles dans un supermarché. Elle est assez facile à mettre en pratique car il suffit d'enrober l'objet porteur de tag d'une feuille d'aluminium pour bloquer les émissions radios des lecteurs.

Les attaques contre les lecteurs RFID sont également à prendre en compte : ajouté à la destruction des lecteurs, qui rend indisponible le processus d'identification de toutes les étiquettes, le vol des lecteurs RFID est aussi un sujet sensible ,un lecteur est réglé sur la fréquence d'émission des étiquettes afin de pouvoir interagir avec ces dernières ,un individu mal intentionné ayant accès à un lecteur peut obtenir frauduleusement des informations critiques qui seraient stockées sur ces étiquettes.

***Contre-mesures**:*
Le contrôle d'accès physique aux étiquettes et aux lecteurs RFID permet de s'assurer qu'ils ne seront pas endommagés ou mis hors-service, si l'étiquette est sous surveillance il est plus difficile pour un attaquant d'y accéder physiquement.

Imitation du lecteur RFID :

En considérant que dans de nombreux cas les communications RFID n'ont pas besoin d'authentification, les attaquants peuvent facilement contrefaire l'identité d'un lecteur légitime afin d'obtenir ou modifier des données présentes sur les étiquettes RFID. La faisabilité de ces attaques repose sur les mesures de sécurité employées pour l'authentification du lecteur RFID. Par exemple, dérober un lecteur RFID qui stocke des informations d'identification, peut permettre d'obtenir ces informations nécessaires pour avoir l'accès aux étiquettes RFID et aux données qu'elles contiennent.

Contre-mesures:

Pour éviter ce genre d'attaques il faut employer un système d'authentification très solide.

Clonage &spoofing (L'usurpation) :

Cloner une étiquette légitime, c'est en produire une copie non autorisée, une personne utilisant cette technique peut avoir accès de manière illégitime à des données, à des objets ou à des lieux. Les étiquettes RFID disposent d'un identifiant unique, dans le cas où une étiquette n'est pas sécurisée, le clonage implique non seulement la copie de l'identifiant de l'étiquette mais aussi de toutes les données associées. Les passeports électroniques allemand sont été sujets à ce type d'attaque.

L'usurpation (*spoofing*) est une variante du clonage, contrairement au clonage, l'usurpation ne reproduit pas physiquement une étiquette RFID pour réaliser ce type d'attaque, l'usurpateur emploie des dispositifs capables d'émuler des étiquettes RFID, en envoyant au lecteur des données imitant celles contenues dans les étiquettes originales. Il peut alors faire passer son étiquette RFID comme valable auprès du lecteur, et obtenir les privilèges associés à l'étiquette usurpée. Cette imitation nécessite un accès complet au même canal de communication que l'étiquette d'origine, pour cela, il est nécessaire d'avoir des connaissances sur les protocoles et secrets utilisés dans l'authentification entre le lecteur et l'étiquette visée.

Contre-mesures :

L'authentification mutuelle, le cryptage et l'utilisation des données complexes sont proposées pour prévenir ce type d'attaques.

Les attaques de spoofing sont généralement évitées en limitant l'accès à l'information"correcte", sans cette information, l'attaque ne peut pas être effectuée. Une clé secrète est donc nécessaire dans le cadre d'une procédure d'authentification, cette clé est ensuite stockée dans une zone restreinte de la mémoire qui ne peut pas être lu et ne sont jamais transmises par le tag en tant que texte. De cette façon, les intrus ne peuvent pas mettre la main sur l'information complète "correcte".

Déni de service (DoS)

Les attaques DoS peuvent prendre différentes formes en attaquant l'étiquette RFID, le réseau ou la base de données, le but est de ne pas voler ou modifier des informations, mais pour désactiver le système RFID de sorte qu'il ne peut pas être utilisé :Lorsqu'un lecteur demande des informations à partir d'une étiquette, il reçoit d'identification et le compare avec l'identifiant stocké dans le serveur de base de données ,le lecteur RFID et le serveur principal sont tous les deux vulnérables aux attaques par déni de service.Lorsque l'attaque DoS a lieu, l'identité n'est pas vérifiée, par conséquent, le service est interrompu.Donc, on doit nous assurer que le lecteur et le serveur de base de données disposent d'un mécanisme pour lutter contre les attaques par déni de service.

deuxattaques de DOS :

a. **Brouillage (Jamming):**

Cette attaque est assez simple a mettre en œuvre, son objectif est de brouiller le signal qui a ciblé les fonctions du système RFID, de sorte que toute communication entre les tags et les lecteurs ne sont pas possibles. Dans les scénarios malveillants, les pirates peuvent utiliser le brouillage technique pour bloquer un propriétaire d'étiquette à l'utiliser aux fins d'identification.

b. **Blindage et cage de faraday :**

L'objectif d'un tel système est de contrôler l'émission et la réception de données, il est possible de s'en servir pour faire du déni de service : Les signaux transmis peuvent être bloqués ou diminués en utilisant un système de cage de faraday, si en plaçant un tag RFID dans une feuille métallique qui peut être en cuivre ou l'aluminium, le lecteur ne peut plus communiquer avec celui-ci .

Contre-mesures :

En général, il est plus facile de détecter les attaques par déni de service que de les empêcher de se produire. Cependant, une fois détectées, les attaques peuvent généralement être arrêtées avant qu'ils font trop de mal.

Ecoute à distance (Eavesdropping ou surveillance non autorisée) :

C'une lune des attaques les plus simple à réaliser au niveau de la couche physique car elle ne requiert que très peu de matériel, elle consiste à écouter une transaction privée à distance entre un lecteur et un tag sans l'accord des participants dans l'intention de révéler des secrets. Les trames enregistrées lors d'une attaque d'écoute à distance vont servir à récupérer des informations plus ou moins importantes comme elles peuvent aussi servir dans le cadre d'une attaque de clonage, spoofing ou replay qui sont souvent précédées d'une attaque eavesdropping permettant à l'attaquant de récupérer les données enregistrées dans la mémoire d'un tag.

Contre-mesures :

Une des façons les plus faciles d'empêcher l'écoute des systèmes RFID est de chiffrer leurs communications avant d'envoyer les données sur la liaison sans fil , de cette façon, lun espionnant peut être en mesure d'entendre la communication, mais pas la déchiffrer.

Skimming ou Activation à distance

Le principe de cette attaque est d'activer et lire une carte sans l'autorisation de son propriétaire, elle permet de communiquer avec la carte en dehors de sa plage de fonctionnement.L'attaquant doit être capable d'alimenter la carte tout en modulant son champ pour envoyer les commandes .Il doit aussi être capable de communiquer avec la carte à une distance supérieure. L'objectif de cette attaque est la récupération des données contenues sur le tag.Les applications sont multiples mais elle est difficile à mettre en œuvre.

Contre-mesures :
Les attaques skimming sont généralement arrêtées par l'utilisation d'un secret partagé, à condition que le lecteur s'authentifie lui-même au tag avant que le tag divulgue l'un de ses données stockées, la lecture non autorisée peut être évitée.

En plus de prévenir les attaques skimming en utilisant des mesures de sécurité sur l'étiquette, elles peuvent également être contrecarrées en utilisant des protocoles bloquant .

II.6.2 Attaques de la couche transport :

L'attaque man-in-the-middle (MITM) :

L'**attaque de l'homme du milieu** (**HDM**) ou *Man-in-the-Middle attack* (**MITM**), parfois appelée **attaque de l'intercepteur**, est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. L'attaquant doit donc être capable de recevoir les messages des deux parties et d'envoyer des réponses à une partie en se faisant passer pour l'autre.[16] [38][39]

Une entité (que nous appellerons « tag truqué ») simule un tag auprès du lecteur légitime et l'autre (que nous appellerons « lecteur truqué ») simule un lecteur auprès du tag légitime. Cette technique permet en quelque sorte de créer une « rallonge » entre le tag légitime et le lecteur légitime, laissant croire à celui-ci qu'il communique directement avec le tag légitime et vice-versa. L'objectif principal d'une attaque MITM est de pouvoir espionner les communications .

Attaque de l'homme du milieu passif : L'attaque de HMP est un cas particulier des attaques de type attaque de l'homme du milieu. Le pirate ne fait que transmettre les messages sous forme passive il ne modifie pas les données.

Contre-mesures:

Plusieurs technologies peuvent être mises en œuvre pour réduire les menaces MITM, telles que le cryptage des communications, l'envoi d'informations via un canal sécurisé, et fournir un protocole d'authentification efficace se basant sur un paradigme d'échange challenge/réponse (défi/réponse), un défi est envoyé au tag qui renvoie une réponse dépendante du défi et pouvant aussi dépendre d'une clé partagée.

Analyse de trafic :

La RFID étant une technologie sans fil, ce type d'attaque est particulièrement redoutable. Rien n'empêche un espion d'intercepter le trafic entre l'étiquette RFID et le lecteur pour en extraire de l'information.

Contre-mesure :

La cryptographie pourrait sembler la solution idéale pour sécuriser les communications entre le lecteur et le tag et diminuer le risque de cette attaque. Avec la cryptographie même si l'intrus analyse le trafic il ne peut pas extraire les informations stratégiques.

II.6.3Attaque de la couche application:

Altération des données :

Bloquer l'émission radio d'une puce RFID est un moyen efficace de dérober un article sans déclencher le système d'alarme qui protège les objets étiquetés. Cependant, pour quelqu'un qui cherche à dérober une quantité importante d'objets, un moyen plus efficace est de changer les données figurant sur les étiquettes attachées aux objets. Selon l'utilisation de l'étiquette des informations comme le prix, les numéros ou d'autres données peuvent être changées. Par exemple, en changeant le prix d'un article et en le passant par une caisse libre service, une personne malintentionnée peut obtenir un rabais spectaculaire sans éveiller le moindre soupçon. Seul un inventaire physique pourra alors révéler une anomalie par rapport à la transaction enregistrée par le système.

Contre-mesures :

Afin de se défendre contre la modification de données sur l'étiquette, le contrôle d'accès à ces étiquettes RFID devrait être pris en compte.

II.6.4 Autres attaques :

Pistage (tracking) :

La fuite d'information et la traçabilité d'une personne ou d'un objet portant un tag, sont deux grandes menaces pour la confidentialité liées à la technologie RFID. En effet, une étiquette peut contenir des données sensibles à propos de la personne qui la porte. Bien qu'utilisé dans beaucoup de domaines à des fins logistiques, comme la gestion des chaînes

d'approvisionnement, le traçage représente une menace contre la confidentialité des données privées. À partir du moment où un objet ou une personne possède une étiquette RFID, il peut être suivi par n'importe quel lecteur capable de lire son étiquette. Ainsi dans les domaines utilisant la RFID à des fins légales, son usage peut être détourné pour réaliser des traçages malveillants des étiquettes et de leurs porteurs[16] [38][39]. Par exemple :

- Un entrepôt peut utiliser la RFID pour savoir quels objets sont présents dans l'inventaire courant, mais un individu peut utiliser le même système pour suivre un objet de l'entrepôt après sa sortie. Cet individu peut ensuite suivre, à l'aide d'une application de traçage non autorisée, les objets de valeurs.

Contre-mesures :

Pour éviter la traçabilité, il faut éviter que l'étiquette communique toute forme d'identifiant à des tiers non autorisés.
Une autre forme de contre-mesure pour la traçabilité est le blindage ,si les étiquettes sont protégées physiquement, elles ne répondent pas aux demandes envoyées par les lecteurs. Ainsi, le lecteur n'a aucun moyen de détecter l'ID, et l'étiquette ne peut pas être tracée.
Une autre méthode simple pour éviter la traçabilité est de désactiver les étiquettes RFID, qui est connu comme «Killing Tag», mais cette méthode peut être elle-même une attaque si elle sera mal utilisées

Canaux de communications

L'attaque dite du **canal auxiliaire** consiste à mesurer les courants et tensions entrants et sortants d'un circuit, en fonction des requêtes qui lui sont envoyées. Par l'analyse poussée de la consommation électrique d'un lecteur d'étiquettes RFID, il est ainsi possible de déterminer la nature de la communication entre le lecteur et l'étiquette, voire dans le cas d'une communication chiffrée, de déterminer la clé de chiffrement.

Une autre attaque sur les canaux de communication, dite du **canal caché**, exploite les cas où l'écriture dans la mémoire de l'étiquette RFID est possible. Un attaquant peut utiliser la mémoire libre de l'étiquette RFID pour y stocker des données non désirées.

Attaques cryptographiques :

Lorsque des informations critiques sont stockées sur les étiquettes RFID, des techniques de chiffrement sont employées afin de préserver l'intégrité et la confidentialité des données protégées. Toutefois, les attaquants déterminés peuvent employer des attaques cryptographiques afin casser les algorithmes de chiffrements utilisés dans les communications entre les étiquettes RFID et les lecteurs ; aux Pays-Bas, une société a montré que la clé

utilisée dans un passeport néerlandais pouvait être facilement cassée à l'aide d'un PC standard au moyen d'une attaque par force brute (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) et en un temps de deux heures[16] [38][39].

Contre-mesures :

Il existe plusieurs mécanisme de défense contre les attaques de cryptographie parmi eux on a :

- ✓ Limitation temporelle des connexions.
- ✓ Augmentation du coût par tentative.
- ✓ Utilisation des protocoles d'authentification de type défi-réponse.

Virus RFID

Comme la plupart des étiquettes RFID passives actuellement ont seulement une petite capacité de stockage de 128 bits, les virus ne sont pas probablement une menace envisageable pour les systèmes RFID. Cependant, les étiquettes RFID peuvent être utilisées comme moyen de transmettre un virus informatique .Il est non seulement possible de lancer des attaques au back-end RFID ou au middleware à partir des étiquettes RFID, mais cela peut être fait même avec des étiquettes de faible coût avec une mémoire seulement capable de stocker 127caractères. Le code malveillant peut prendre la forme à la fois d'un ver et d'un virus, et peut donc se propager soit par les connexions du réseau ou par le moyen du système RFID lui même.

Contre-mesures :

Actuellement, toutes les attaques qui ont été provoquées par des virus RFID tel que les attaques de dépassement de mémoire tampon ou attaques par injection SQL sont toutes connues et il y'a aussi des contre-mesures bien déterminées pour les combattre comme la vérification des limites, la vérification des paramètres de la liaison sans fil, limiter les autorisations d'accès à la base de données.

II.7. Conclusion :

Les problèmes de sécurité dans les systèmes RFID devient de plus en plus incontournable, pour remédier aux problèmes de sécurité plusieurs solutions ont été déjà proposées.
Dans ce chapitre nous avons donné une description détaillée sur les mécanismes de sécurité tel que le chiffrement des données et les signatures numériques, leurs principes de fonctionnement et leurs utilisations puis nous avons essayé de découvrir certaines attaques possibles qui peuvent affecter les systèmes RFID et nous avons discuté des mesures de sécurité possibles qui peuvent être utilisés pour lutter contre ces attaques.

Le chapitre suivant prend la charge la présentation de quelques protocoles cryptographiques des systèmes RFID déjà mises en place.

III .1. Introduction :

Les techniques RFID sont en proie à la sécurité et problèmes de confidentialité dus au canal de communication sans fil et insécurisé, afin d'aborder les questions de sécurité des systèmes RFID, différentes solutions ont été proposée. Parmi ces dernières, les protocoles cryptographiques qui représentent la clé de base de la sécurité RFID.

Dans ce chapitre, Dans un premier temps nous allons présenter quelque protocole d'authentification dans les systèmes RFID, Ensuite, nous proposons un protocole d'authentification qui utilise des générateurs de nombres pseudo-aléatoires (PRNG) et quelques opérations cryptographiques simples.

III.2. Les protocoles cryptographiques dans les systèmes RFID :

Les protocoles cryptographiques sont des petits programmes qui spécifient une séquence d'émissions/réceptions de messages qui visent à établir entre deux ou plusieurs participants (agents), des communications répondant à certaines propriétés de sécurité telles que la confidentialité, l'intégrité, l'authentification, ces protocoles utilisent des primitives cryptographiques à faible coût comme le bitwise opérations, générateurs de nombres pseudo-aléatoires, fonctions de hachage, etc .
Deux types de protocoles cryptographiques existant: légers (lightweight) et ultralégers (Ultralightweight) qui attirent beaucoup d'attention car ils sont plus adaptés pour les limites de ressources d'étiquettes RFID.

III.2.1 Les protocoles lightweight :

Ce type de Protocoles sont très utilises dans l'industrie du a leur avantage de maintenir la demande de calcul et le coût très faible des étiquettes RFID.La cryptographie lightweight est divisée en deux :

Primitives lightweight: cette catégorie regroupe les primitives symétriques, les primitives asymétriques, les fonctions de hachage et les générateurs de nombres aléatoires.

Protocoles lightweight: utilisent les primitives lightweight pour fournir des propriétés de sécurité. Cette catégorie peut être divisée en cinq sous- catégories: les protocoles d'identification, les protocoles d'authentification, protocoles de distance bounding, les protocoles de regroupement de preuve et protocoles de propriété du tag .

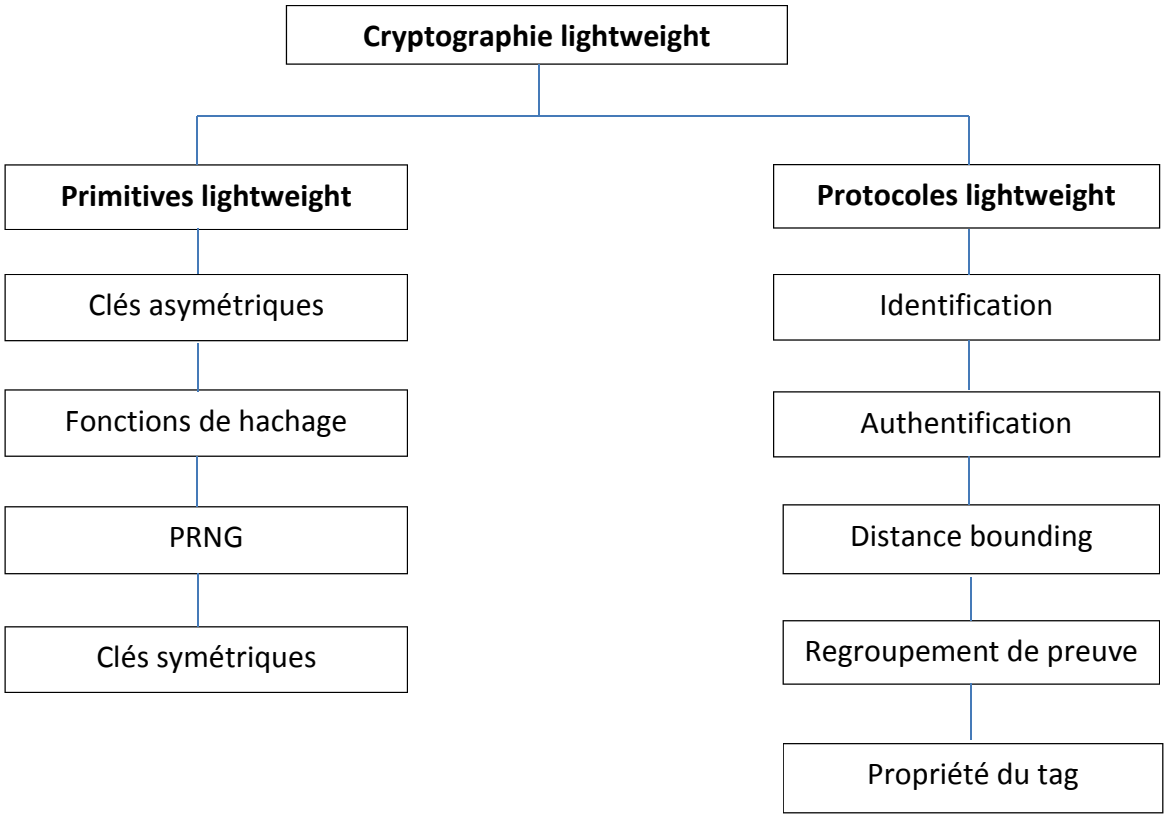


Figure III.1: Les protocoles légers dans les systèmes RFID.

III.2.2 Les protocoles ultralightweight

Ce sont des protocoles basés sur les opérations ultralégères visent à fournir une authentification sans utiliser des primitives cryptographiques et entraîner uniquement des opérations binaires simples et arithmétiques sur le tag (par exemple XOR, AND, OR, rotation, etc.).

Les protocoles ultralightweight sont divisés en deux groupes principaux: les protocoles basés sur "minimaliste cryptographie" et des protocoles basés sur des problèmes mathématiques NP-difficiles.III.3 Présentation de quelques protocoles d'authentification dans les systèmes RFID :

Au cours des dernières années, de nombreux protocoles d'authentification ont été proposes afin de surmonter les problèmes des systèmes RFID, on présente ici quelque uns :

III.3.1 Le protocole Fan et al :

Fan et Al ont propose un Protocol d'authentification mutuelle léger. dans ce protocole les trois composants du protocole pré-partagent le tuple (K_i, cro (·), Rot (·), PRNG(·))

Liste des notations utilisées :

La liste des notations utilisées dans le protocole fan et al est la suivante :

Notations	description
RID	Identifiant prive du tag
TID	Identifiant prive du lecteur
Nr , Nt, Ns	Numéro aléatoire généré par le lecteur l'étiquette et le serveur respectivement
Ki	La clé de la i-éme session
PRNG(.)	La fonction Pseudo Random Number Generator
Cro(x,y)	Le fonctionnement du bit cross
Rot(x,y	L'opération de rotation, x=W(y)
⊕	Opération OU exclusif.
Mark	Le statut de la dernière session.
||	Opération de concaténation.

Tableau III.1: liste des notations utilisées dans le protocole Fan et Al.

Tableau de données d'index :

La table de données d'index incluait la valeur d'index et l'index contenu qui sont uniques .Dans chaque session, la valeur de la clé est mise à jour, donc la valeur de l'index est fraîche pour chaque session. De plus, après chaque session réussie, le statut de Mark passe de "00" à "10".

Valeur d'index	Contenu d'index
cro(RID_TID ; K1)	Rot(K1_TID ; K1_RID)
cro(RID_TID ; K2)	Rot(K2_TID ; K2_RID)
: : :	: : :
cro(RID_TID ; Ki)	Rot(Ki_TID ; Ki_RID)
cro(RID_TID ; Ki+1)	Rot(Ki+1_TID ; Ki+1_RID)

Tableau III.2: table dindex du protocole Fan et Al.

Description du protocole :

Le protocole fan et al, comme indiqué sur la figure (), exécute les étapes suivantes :

tape1 :

Le lecteur démarre le protocole en envoyant un nombre aléatoire N_R à l'étiquette.

etape2 :

Après la réception de N_R , l'étiquette génère un nombre aléatoire N_t et place Mark = 00.

Elle transmet ensuite cro (RID $\oplus$ TID, K_i) , N_t au lecteur.

Etape3 :

Après avoir reçu le message, le lecteur obtient N_t et envoie cro (RID $\oplus$ TID, K_i) , N_t , N_R au serveur.

Etape4 :

Le serveur reçois N_R et N_t utilise cro (RID $\oplus$ TID, K_i pour trouver le contenu d'index correspondant dans l'IDT.

S'il peut trouver une correspondance, cela indique que la dernière session a été faite correctement et la session en cours est exécutable.

Il génère un nombre aléatoire N ss
envoie cro (RID $\oplus$ TID, $K_i \oplus N_S$) Rot ($K_i \oplus$ TID, $K_i \oplus$ RID) $N_S \oplus K_i$ au lecteur.

Sinon, le l'authentification échoue et le protocole sera terminé

Etape 5 :

Une fois que le lecteur a reçu le tuple (cro (RID $\oplus$ TID, $K_i \oplus N_S$) Rot ($K_i \oplus$ IDD, $K_i \oplus$ RID) $N_S \oplus K_i$), en fonction du poids W ($K_i \oplus$ TID) de l'opération de rotation et $K_i \oplus K_i \oplus$ TID il obtient TID.

Il obtient alors N_S et vérifie la valeur de cro (RID $\oplus$ TID, $K_i \oplus N_S$) en comparant avec la valeur reçue.

Si c'est le cas, il calcule TID $\oplus N_R$ et TID $\oplus N_S$ et les envoie à l'étiquette.

Etape 6 :

Après avoir reçu ce message, l'étiquette obtient N_S et

Si TID = TID $\oplus N_R \oplus N_R$ tient, il authentifie le serveur et le lecteur.

Alors l'étiquette met à jour K i comme K_{i+1}= cro ($N_R \oplus N_S \oplus N_t$, K_i) et l'envoie au lecteur impliqué dans le message cro (RID $\oplus$ TID, K_{i+1}).

Sinon, l'authentification échoue.

Etape 7 :

Lors de la réception du message cro (RID $\oplus$ TID, K_{i+1}),

si cro (RID $\oplus$ TID, cro ($N_R \oplus N_S \oplus N_t$, K_i)) = cro (RID $\oplus$ TID, K_{i+1})

le lecteur met à jour K_i par la même équation K_{i+1} = cro ($N_R \oplus N_S \oplus N_t$, K_i) et l'envoie au serveur par le message cro (RID $\oplus$ TID, K_{i+1}).

Sinon, le protocole sera terminé.

Etape 8 :

Une fois que le serveur a reçu ce message, il effectue la même opération de vérification et s'il est en attente, le serveur met à jour K_i comme K_{i+1} = cro ($N_R \oplus N_S \oplus N_t$, K_i). Il calcule ensuite le message K_{i+1} TN T $\oplus N_R$ et l'envoie à le lecteur. Sinon, la connexion échoue

Etape9 :

A la réception du message $K_{i+1} \oplus N_t \oplus N_R$, si K_{i+1} = ($K_{i+1} \oplus N_t \oplus N_R$) $\oplus N_t \oplus N_R$ tient, le lecteur vérifie K_{i+1} et envoie le message $K_{i+1} \oplus N_t \oplus N_R$ à l'étiquette pour le même processus de vérification.

Autrement le protocole sera terminé.

etape10 :

Une fois que l'étiquette accepte la validité de K_{i+1} , elle marque Mark = 01, indiquant que la synchronisation de K_i est com- terminé.

Ensuite, l'étiquette calcule Mark $\oplus N_S$ et l'envoie au serveur via le lecteur. Notez que dans le travail original [2], Mark a été défini comme une chaîne de 2 bits alors que les paramètres comme N_S sont généralement plus grands chaînes, donc leur XOR n'a pas de sens. Cependant, sans perte de généralité, nous supposons que Mark est une extension triviale de cette chaîne de 2 bits.

etape11 :

Après avoir reçu le message Mark $\oplus N_S$, le serveur obtient la valeur de Mark et si elle est égale à 01, il conclut que la synchronisation de K i est terminée. Ensuite, le serveur ajoute un nouveau record cro (RID $\oplus$ TID, K_{+1i}), Rot ($K_{i+1} \oplus$ TID, $K_{i+1} \oplus$ RID) à IDT, après quoi la notification que l'enregistrement est complet la mise à jour est envoyée à l'étiquette via le lecteur.

etape12 :

Maintenant, l'étiquette définit Mark = 10, indiquant que le protocole d'authentification est terminé.

III.3.2 Le protocole HMNB :

Est un protocole d'authentification pour les RFID, il utilise une primitive cryptographique : la fonction de hachage [51] [52] [53]

Liste des notations utilisées :

La liste des notations utilisées dans le protocole fan et al est la suivante :

Notations	description
R, T	R : lecteur, T : étiquette.
TID	Identifiant prive du lecteur.
Nr, Nt	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
H	fonction de hachage.
ID	identificateur partage entre le lecteur et l'étiquette.
||	Opération de concaténation.
IDP	L'ancienne valeur de l'ID.
HID	Fonction de hachage de l'ID.

Tableau III.3: liste des notations utilisées dans le protocole HMNB.

Description du protocole

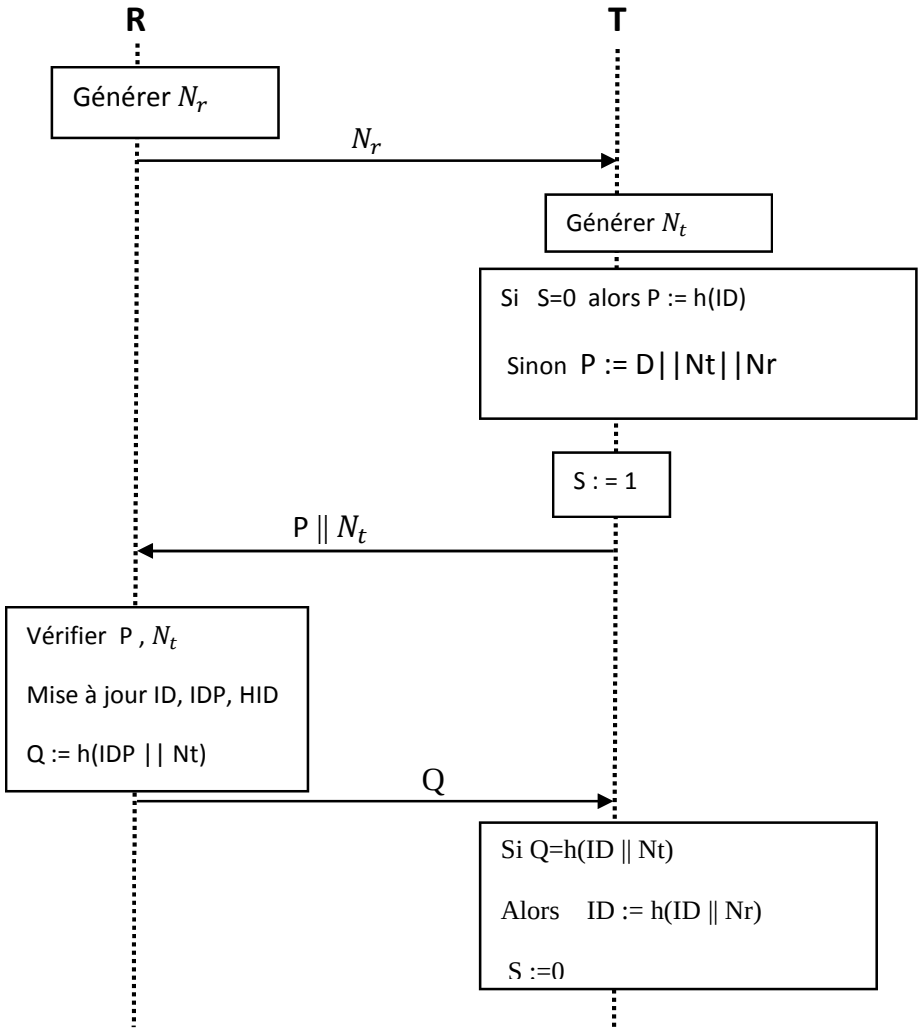


Figure III.3:schéma général du protocole HMNB

Etape 1 :

Le protocole est lancé par le lecteur : le lecteur génère un nombre aléatoire Nr et l'envoie a l'étiquette.

Etape 2 :

Apres avoir reçu Nr l'étiquette génère un nombre aléatoire Nt. Puis vérifie son état :

Si S=0 alors elle calcule P := h(ID)

Sinon calcule Sinon P := h(ID) || N_t || N_r

puis met son état a 1 (S = 1) et envoie P et Nt a l'étiquette.

Etape 3 :

A la réception de N_t et P le lecteur les vérifies puis

Il mets a jours ID, IDP, HID

Calcule Q := h(IDP || Nt)

Envoie Q a l'étiquette.

Etape 4 :

L'étiquette reçois Q et vérifie Q = H(ID||Nt), si oui T met à jour son ID puis remet son état à 0.

III.3.3 Le protocole RDW :

Liste des notations utilisées :

La liste des notations utilisées dans le protocole RDW est la suivante [51] [52] [53] :

Notations	description
R	R : lecteur,
T	T : étiquette.
Nr, Nt	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
K	Clé unique et partagée

Tableau III.4: liste des notations utilisées dans le protocole FDW.

Description du Protocol

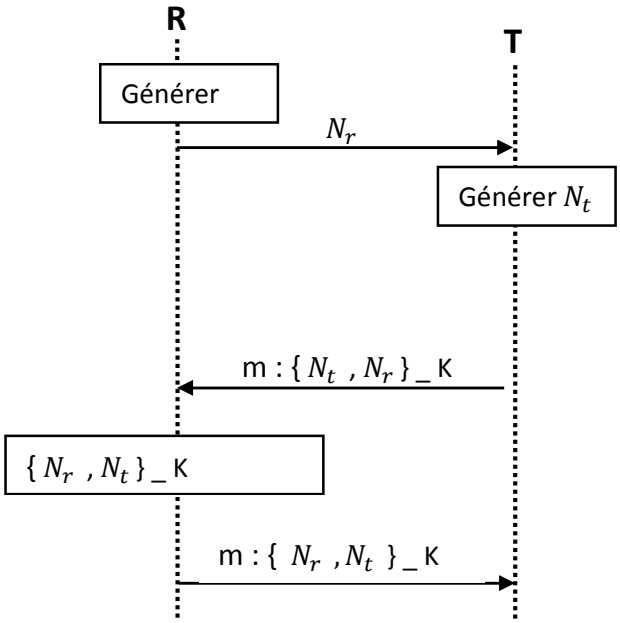


Figure III.4:Schéma général du protocole RDW

Dans ce protocole, chaque couple de lecteur R et tag T possède une clé unique et partagée K.

Etape 1 :

Le lecteur lance le protocole par l'envoi d'un nonce Nr au tag.

Etape 2 :

Le tag génère un nombre nonce Nt et crypte la paire (Nt, Nr) avec la clé partagée K, et l'envoie au lecteur.

Etape 3 :

Le lecteur déchiffre le message en utilisant la même clé partagée et inverse l'ordre des deux nonces, crypte le message avec la même clé partagée et l'envoie au tag.

La notation Alice-Bob pur ce protocole est:
R → T : Nr
T → R : {Nt,Nr}_K
R → T : {Nr,Nt}_K

III.3.4 Le protocole R^2AP (Reconstruction based RFID Authentiquassions Protocol) :

Est un protocole RFID ultraléger basé sur l'utilisation d'une nouvelle opération de reconstruction au niveau du bit. On note que la nomination des protocoles revient aux premiers caractères de nom des auteurs[50] .

Liste des notations utilisées :

Les notations utilisées dans le protocole R^2AP sont présentées dans le tableau suivant :

Notations	description
R	R : lecteur,
T	T : étiquette.
ID	Identificateur partagé entre le tag et le lecteur
IDS	Index de la table ou elle est stocké les secrets du tag
K1, K2, K3	Clés symétriques partagées entre le tag et le lecteur
N1, N2	Nombres aléatoires généré par le lecteur
Rot(x,y)	Rotation gauche de x par y bits
Rec(x,y)	L'opération de reconstruction de x par y
||	Opération de concaténation

Tableau III.5: liste des notations utilisées dans le protocole R^2AP.

Description du protocole

Ce protocole est basé sur l'échange des IDS, ID et les trois clés secrètes, désignées par K1,K2 et K3, et toutes les chaînes utilisées dans ce protocole doivent avoir une longueur l

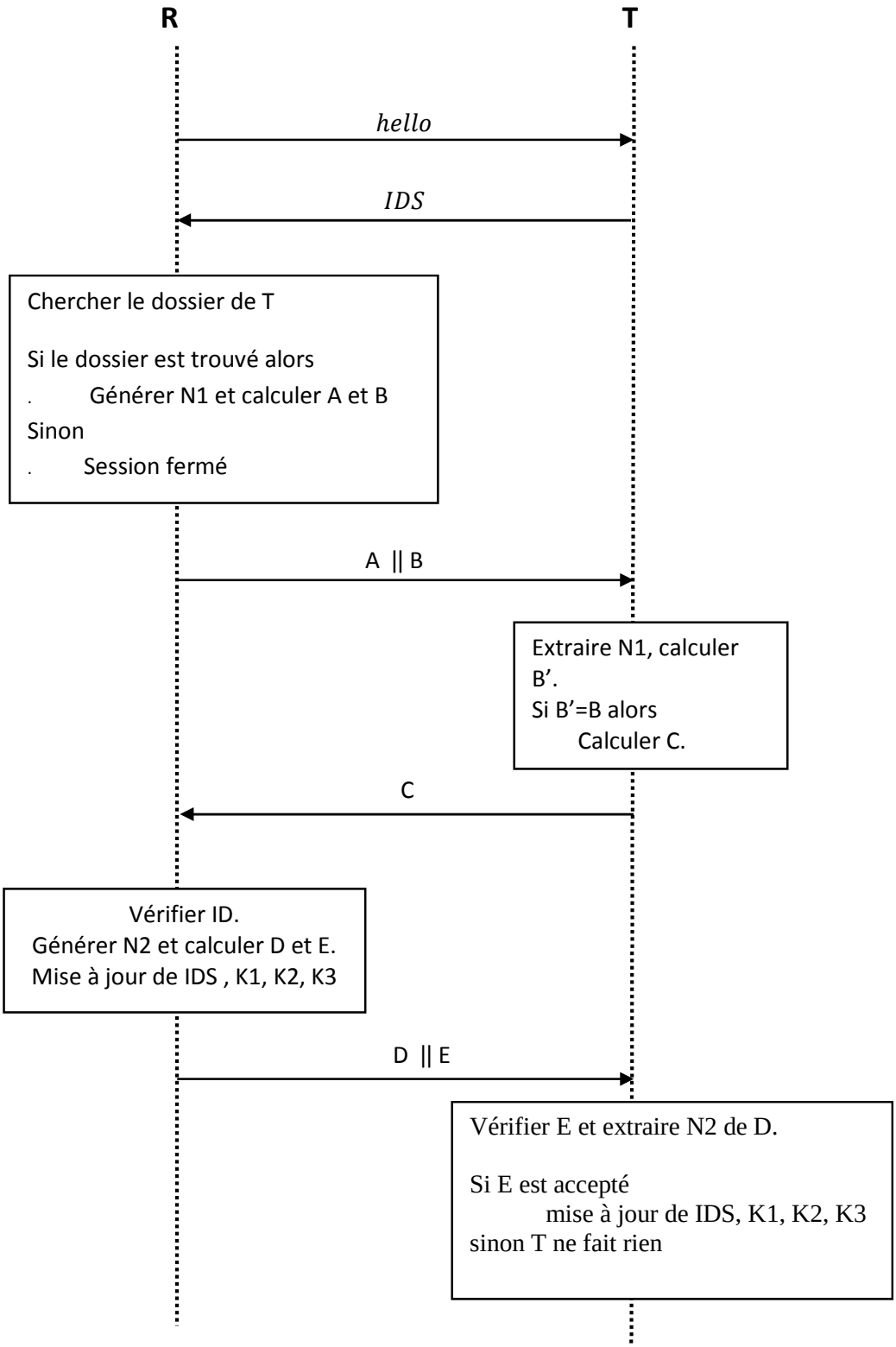


Figure III.5:schéma général du protocole R^2AP

Étape 1:
Le lecteur R envoie un message Hello au tag T pour initialiser une nouvelle session de protocole.

Étape 2:
T répond à R avec ses IDS.

Étape 3:
À la réception des IDS de T, R les utilisent comme un index de recherche des secrets du tag dans la base de données.
Si R trouve le dossier, l'étape 4 sera effectuée, autrement, R met fin à la session actuelle de protocole.

Étape 4:
R génère un nombre aléatoire N1, puis transmet des messages A et B à T, où

$A = Rec (K1 || K2) \oplus N1$ et
$B = Rot (Rec (K2 || N1), Rec (K3 || N1)) \oplus Rot (N1 || N1)$.

Étape 5:
Après avoir reçu les messages A et B, T extrait le nombre aléatoire N1 de message A et calcule ensuite un message B' en utilisant la formule de B avec les clés secrètes K1, K2, K3 et le nombre aléatoire N1 extrait.
Si B = B', T authentifie R comme un lecteur valide et transmet ensuite le message C comme une réponse, où: $C = Rec (Rec (K2 || K3), Rec (N1 || K1)) \oplus ID$.
Sinon, T met fin au protocole.

Étape 6:
Après avoir reçu C, le lecteur R peut authentifier T en utilisant l'ID extraite du message C.
Si l'ID correspond à celui de la base de données principale, R génère un nombre aléatoire N2 et calcule les messages D et E comme suit, et les transmet à T:

$D = R (N1 || K3) \oplus Rec (K1 || K3) \oplus N2$ et
$E = Rot (Rec (K2 || N2), Rec (K2 || N1)) \oplus Rot (N2 || N2)$.

Et puis, R mettra à jour ses secrets comme suit:
$IDSnew = Rec (I DS \oplus N2 || K3) \oplus K1$
$K1new = Rec (N2 || N1) \oplus K2$
$K2new = Rec (K2 || N1 \oplus N2) \oplus K3$
$K3new = Rec (K2 || K3) \oplus N1$

Étape 7:
Lors de la réception des messages D et E, T extrait le nombre aléatoire N2 du message D et teste la validité du message de E en utilisant ses clés secrètes. Si T accepte E, elle mit à jour ses secrets de la même manière que R. Sinon, T ne fait rien.

La notation Alice-Bob pour ce protocole est:

R → T: Hello
T → R: IDS
R → T: A || B
T → R: C
R T: D || E % tel que :
IDSnew = Rec (IDS ⊕ N2 || K3) ⊕ K1
K1new =Rec (N2 || N1) ⊕ K2
K2new = Rec (K2 || N1⊕ N2) ⊕ K3
K3new = Rec (K2 || K3) ⊕ N1

III.3.5 Le protocole EKE :

Liste des notations utilisées :

La liste des notations utilisées dans le protocole EKE est la suivante :

Notations	description
R, T	R : lecteur, T : étiquette.
Na , Ea, Nb	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
Kab	clé symétrique
K	clé de session

Tableau III.5: liste des notations utilisées dans le protocole EKE

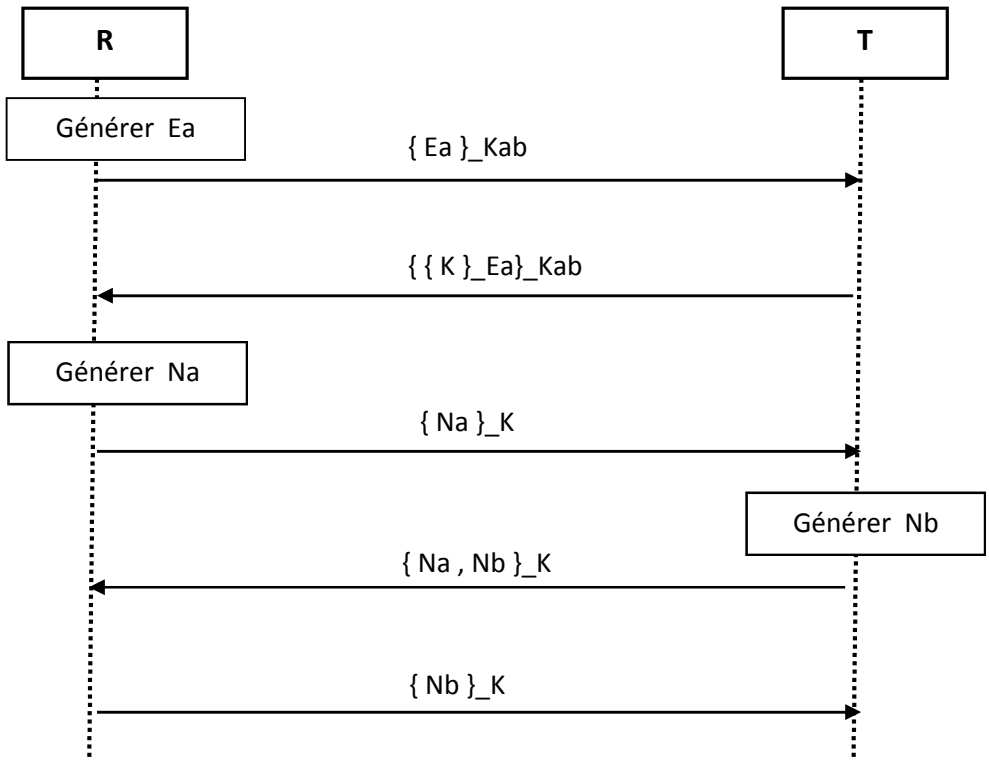


Figure III.6: le protocole EKE

La notation Alice-Bob pour ce protocole est:

A ⟶ B : { Ea }_Kab

B ⟶ A : { { K }_Ea}_Kab

A ⟶ B : { Na }_K

B ⟶ A : { Na , Nb }_K

A ⟶ B : { Nb }_K

Le protocole EKE, comme indiqué sur la figure (), exécute les étapes suivantes :

Etape 1 :
Le lecteur démarre le protocole en envoyant { Ea }_Kab (un nombre aléatoire E_a cripté par la cle symétrique Kab) à l'étiquette.

Etape 2 :
l'étiquette décrypte le message reçu pour obtenir Ea puis envois { { K }_Ea}_Kab (Ea chiffre par la clé de session K et la clé symétrique Kab.

Etape 3 :
Le lecteur envoie { Na }_K (un nombre aléatoire Na crypté avec la clé de session K) a l'étiquette

Etape 4 :

l'étiquette décrypte le message pour obtenir Na puis généré un autre nombre aléatoire Nb et envoi { Na , Nb }_K au lecteur

Etape 5 :
Le lecteur décrypte le message pour obtenir le Na et le compare au Na déjà généré : si ils correspondent donc l'étiquette est authentifié
Puis envoie { Nb }_K a l'étiquette

Etape 6 :
L'étiquette décrypte le message pour obtenir Nb puis le compare au Nb déjà généré si ils correspondent donc le lecteur est authentifié.

III.3.6. Schéma d'authentification proposé :

Nous proposons ici notre schéma de sécurisation d'un système RFID. Il utilise des opérations cryptographiques légères comme les PRNG, fonctions de hachage et XOR,
Dans notre schéma, nous utilisons des nombres aléatoires avec la valeur secrète de tag comme
Ensemencer et mettre à jour le secret de la balise après chaque authentification. La propriété de hasard des générateurs aide assurer la confidentialité et l'immunité contre les attaques par répétition 8 . Comme le secret est mis à jour après chaque session, transférez le secret est également assuré.

Hypothèse :

Nous supposons le canal de communication entre lecture et le serveur principal est entièrement sécurisé. Nous concentrons alors sur la sécurité de canal de communication entre le tag et le lecteur.
Le tag est un périphérique passif et communique avec le lecteur via un canal non sécurisé.
L'étiquette contient deux champs de données :

S : le secret de l'étiquette il est de 128 bits
ID : le pseudonyme d'étiquette (valeur d'index dans la base de données) est de 96 bits

Le serveur principal contient une base de données locale contenant des champs:

ID, h (ID) , $S_{nouveau}$

Liste des notations utilisées :

Notations	description
S,ID	Secret et ID pseudonyme de tag.
$S_{nouveau}$	Secrets de session actuels de la balise stockée dans le serveur principal
h ()	fonction de hachage.
PRNG(A)	pour trouver un nombre aléatoire avec A comme valeur de départ.
PRNG(A,B)	pour trouver un B.Nombre pseudo_aléatoire avec A comme valeur de départ.
||	Fonction de concaténation de chaines.
⊕	La fonction XOR.

Tableau III.6: liste des notations utilisées dans le protocole proposé

Description du Protocole :

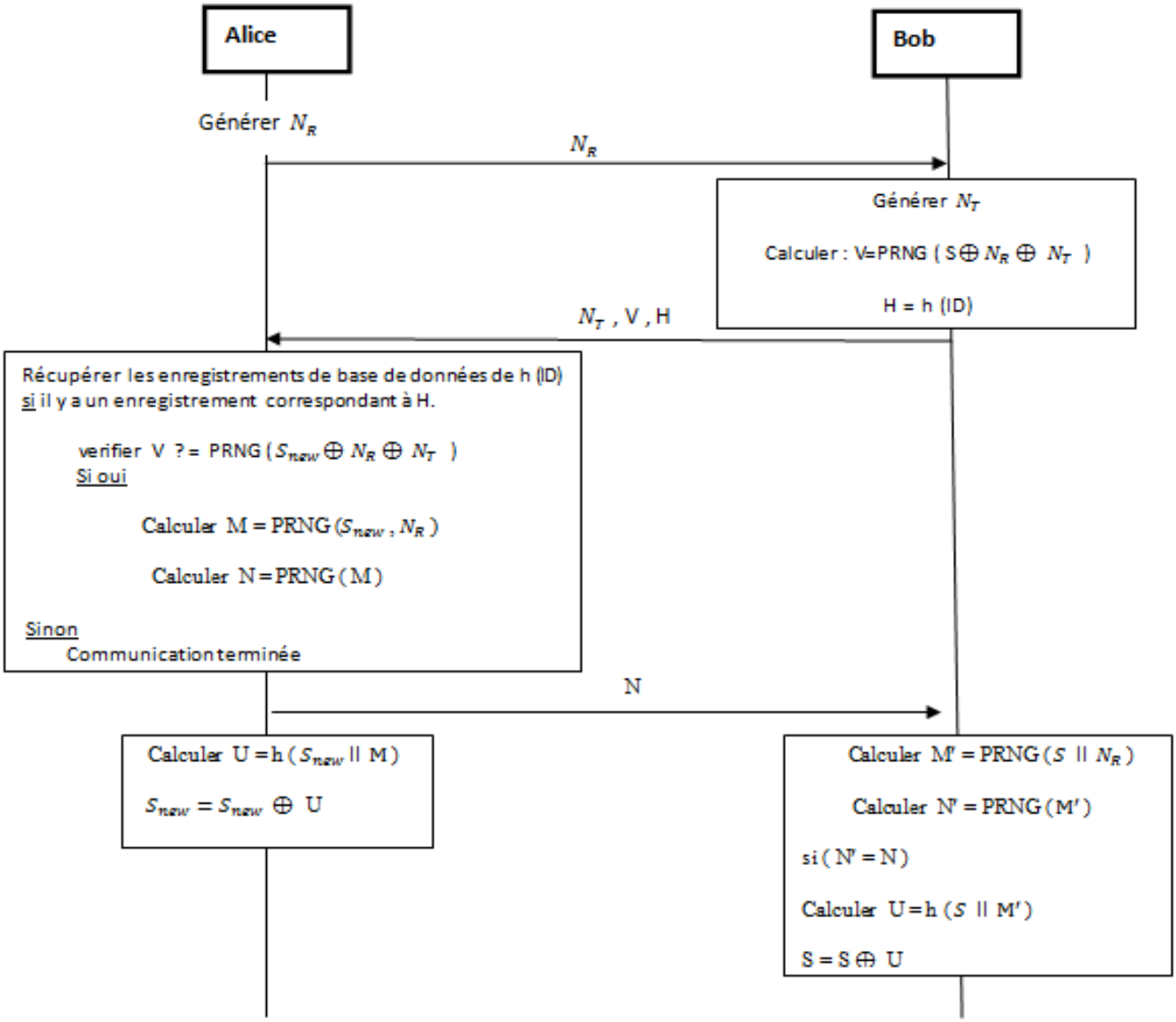


Figure III.7: Schéma générale de protocole proposé

Ce protocole exécute les étapes suivantes :

Étape 1:
Le lecteur initie le processus d'authentification en interrogeant le tag puis sélectionne un tag et demande ses informations.
Le lecteur génère un nombre aléatoire (N_r) et l'envoie à l'étiquette .

Étape 2:
Après avoir reçu $N_r$du lecteur, l'étiquette génère un autre nombre aléatoire (N_t). puis calcule V = PRNG (S $\oplus N_R \oplus N_t$).) et H = h (ID), en utilisant les informations (Tag ID, S) stockées dans l'étiquette.
L'étiquette envoie V, H et N_t au lecteur.

Étape 3:
Le lecteur après avoir reçu ces valeurs, récupère les enregistrements de base de données de h (ID) pour trouver s'il y a un enregistrement correspondant à H.

• Si aucun enregistrement ne correspond à cette valeur, la communication est terminée.
• Si un enregistrement est trouvé, le serveur extrait le secret S New de la balise

correspondante de la base de données et calcule V ' = PRNG (S Nouveau $\oplus N_r \oplus N_t$) pour vérifier si V ' et le V reçu sont identiques ou non.

Si elles sont égales,c'est confirmé que la session précédente a été couronnée de succès et tag contient S new comme valeur S.

Étape 4:
En utilisant M comme valeur de départ, il calcule un nombre aléatoire (N) à partir de PRNG.
Le lecteur envoie N au l'etiquette.
La même chose doit être faite au niveau de l'étiquette

Étape 5:
Pour confirmer que le lecteur est authentifié, l'étiquette calcule M ' =PRNG(S || N_r) le nombre pseudo-aléatoire en prenant S (secret stocké dans l'étiquette) comme valeur initial.

En utilisant M 'comme valeur de départ, il calcule un nombre aléatoire (N ') à partir de PRNG.
La balise vérifie si N ' et N sont identiques. Si oui, l'étiquette est confirmée que l'information provient du lecteur légitime. L'étiquette calcule U = h (S || M ') et met à jour sa valeur secrète en tant que S⊕U.

Étape 6:
Le lecteur calcule U = h (S New || M).
Puis met à jour la valeur secrète dans la base de S new = S new ⊕ U.

III.4. Conclusion :

Ce chapitre est consacré aux protocoles cryptographiques dans les systèmes RFIDs, dans on a présente quelques protocoles d'authentification et dans le chapiste suivant on verra la vérification automatique de la sécurité des protocoles cryptographiques avant leurs mises en service en utilisant des outils automatisés tout en concentrons sur l'outil AVISPA qui permet la spécification, l'analyse et la validation des protocoles de sécurité

Chapitre IV :

La vérification formelle d'un protocole RFID a laide D'AVISPA et SPAN

IV.1 Introduction :

La vérification de la sécurité des protocoles cryptographiques avant leurs mises en service est très importante car leur sécurité n'est pas garantie par l'usage des méthodes de chiffrement seulement, mais aussi par une vérification automatique.

Nous présentons donc dans ce chapitre quelques notions relatives à la vérification, en se basant sur le modèle formelle, ces caractéristiques, ces méthodes et les outils de vérification automatique qui s'appuient sur ce modèle. Nous concentrons sur l'outil AVISPA qui permet la spécification, l'analyse et la validation des protocoles de sécurité.

Ensuite on verra le langage formel HLPSL (High Level Protocol Specification Language) qui permet la spécification modulaire afin de vérifier les propriétés de sécurité propose dans le chapitre 3 à l'aide de l'outil AVISPA .

IV.2 Les notions de base de la vérification :

La vérification est l'étape qui permet de tester, de prouver et de confirmer qu'un protocole est sûr ou non :

Les protocoles : Les protocoles cryptographiques sont des petits programmes qui spécifient une séquence d'émissions/réceptions de messages qui visent à établir entre deux ou plusieurs participants (agents), des communications répondant à certaines propriétés de sécurité

Les agents : Il y a deux types d'agents ou participants :

Participants honnêtes *:*
Ce sont Les agents qui ont un comportement qui suit celui énoncé par une exécution normale du protocole

Intrus :
Un intrus ou attaquant est un participant qui ne suit pas exactement le déroulement du protocole. Il espionne les communications qui circulent entres les agents, joue plusieurs sessions de protocoles avec des participants, en se faisant passer pour un agent honnête et ainsi effectue des actions non prévues par la spécification du protocole, afin de découvrir des informations supposées rester secrètes.

L'attaque :
Une attaque est l'exécution d'une ou plusieurs sessions du protocole qui permet à l'intrus d'apprendre une information supposée secrète. On peut modéliser une attaque en utilisant un outil automatique de vérification de protocoles cryptographiques comme AVISPA .

Secret :
Un secret est une donnée confidentielle ne devant pas être découverte par une tierce personne qui n'est pas censée la connaitre. Un protocole vérifie la propriété de secret, si une personne malhonnête (l'intrus) en fonction de ses capacités ne peut jamais obtenir les données échangées entre plusieurs participants honnêtes.

IV .3 Les objectifs de la vérification :

La vérification de la sécurité des protocoles cryptographiques dépend généralement de deux axes complémentaires : la recherche d'une attaque et la preuve d'un protocole sûr [57] .

IV.4 La vérification formelle :

La vérification formelle est une méthode qui permet la description mathématique d'un protocole et les propriétés de sécurité, ainsi les étapes à suivre pour déterminer si le protocole satisfait ces propriétés (si le protocole est sécurisé) [58].

Avant de faire l'étude des protocoles de sécurités, il est nécessaire de les modéliser, à l'aide de cette modélisation on feras une vérification formelle qui nous permet de conclure si le protocole ne révèle en aucun cas une certaine faille ou qu' il assure certaines propriétés.

A la fin de la vérification la détection des attaques est possible ou, au moins, des points faibles que la spécification d'un protocole peut comporter.

L'objectif de modèle symbolique est le fait qu'il permet d'obtenir des preuves mathématiques simplifiées et souvent automatisées.

IV.4.1 Les concepts de base relatifs aux modèles symboliques utilisés dans ce mémoire :

Les messages :

Dans les modèles symboliques, les messages sont généralement représentés par des termes, pour cette raison, la définition de modèles symboliques passe par la définition d'une algèbre de termes. Par exemple, si k et m sont des termes, le chiffrement symétrique de m par k est noté par {m}_k ,Les primitives cryptographiques sont donc représentées par des symboles.

Les primitives cryptographiques :

Les primitives cryptographiques comme la fonction de hachage et les clés asymétriques, sont considérées comme des boîtes noires, c'est-à-dire qu'elles sont sûres par définition et qu'il n'existe pas d'algorithme pour les casser.

Les systèmes de transition :

L'exécution d'un protocole est modélisée par une séquence finie d'états globaux et des transitions entre ces états globaux, un état global contient tout les informations sur les messages échangés et les états locaux des agents. L'état local d'un agent est comparable à l'état de son mémoire à un certain moment d'exécution du protocole. Nous obtenons ainsi une trace d'exécution d'un protocole qui est une suite alternante d'états globaux et de transitions entre ceux-ci.

Le canal de communication :

Le but de canal de communication sert à l'échange des messages entre les agents. On distingue deux types de canaux de communications : les canaux de communications publics et les canaux de communications privés :

Les canaux de communication publics : un canal de communication est dit public si Les messages échangés sur ce canal peuvent être vus par tous les participants qu'il soit honnête ou non c'est a dire que ces messages sont connus de tous, en particulier de l'intrus qui est capable d'initialiser le protocole avec d'autres participants, il peut intercepter, rejouer ou modifier les messages pendant l'exécution.

Les canaux de communication privés : dans ce type de de canaux de communication seuls ces participants honnêtes peuvent recevoir et envoyer des messages. Par conséquent, un intrus ne peut pas donc écouter les messages qui circulent sur ce genre de canaux.

IV.4.2 Les méthodes de vérification formelle :

Il existe trois principales méthodes de la vérification formelle de protocoles : La méthode du raisonnement logique, les méthodes d'exploration de l'état et les méthodes de démonstration de théorèmes[59] .

La méthode du raisonnement logique :
C'est la première méthode utilisé dans la vérification automatique des protocoles parmi ses techniques essentielles on cite :

- ***La BAN*** : C'est la plus importante technique de raisonnement, elle décrit les principes de communication, messages envoyés et reçus tout au long du processus d'exécution du protocole. Ces raisonnements logiques aboutissent à une assertion finale qui permet de juger si un protocole est correct ou pas.

- ***La GNY (extension de la BAN)***.Plus sophistiquée et complexe, elle améliore la BAN en mettant en évidence la différence entre le contenu d'un message et sa signification .

- ***La BGNY (extension du GNY)*** : Elle permet de spécifier les propriétés du protocole à des niveaux intermédiaires (insertion de l'opération de hachage, algorithmes d'échange de clés…)

L'approche par preuve de théorèmes
Cette approche est basée sur un raisonnement mathématique pour démontrer l'exactitude d'un protocole de sécurité, Les outils développés pour cette méthode comme ACL2, Isabelle, utilisent respectivement la FOL (First Order Method) et la HOL (High Order Method). Cette approche permet d'obtenir une preuve formelle, mais n'est pas totalement automatisée
Les méthodes d'exploration d'états
Ces méthodes basées sur la construction d'un modèle de protocole et de vérifier que chaque état atteignable satisfait certaines propriétés.
Ce type de méthodes peuvent être divisées en deux catégories :

La vérification non-bornée : Dans cette approche, on recherche toutes les exécutions

possibles du protocole, en vérifiant chaque état accessible qui satisfait certaines conditions, ,elle considère une infinité de principes et une infinité de sessions. L'espace des états généré par l'analyse d'un protocole peut être infini, on parle alors d'explosion d'états.

La vérification bornée : Dans cette méthode le nombre de sessions est fixé, La méthode de vérification recherche alors un état particulier où les propriétés de sécurité sont violées et génère un contre-exemple en exploitant une trace depuis cet état jusqu'à un état initial. C'est sur cette approche que reposent tous les outils de vérification automatique de protocoles.

IV.4.3 outils de vérification formelle :

Dans le domaine de vérification automatique des protocoles de sécurité, Il y a plusieurs analyseurs de protocoles ,mais la plateforme AVISPA (Automated Validation of Internet Security Protocols and Applications) est l'analyseur le plus connu qui modélise un grand nombre de protocoles. L'efficacité d'Avispa a été testée sur de nombreux protocoles récemment standardisés, par exemple par l'IETF (Internet Engineering Task Force) et des protocoles du domaine e-business.
Cet outil est compatible avec loutil graphyqique SPAN et possède une syntaxe simple par rapport aux autre outils . pour ces raison nous l'avons utilisé dans notre projet[59] .

La plateforme AVISPA :

En juillet 2005 les partenaires du projet européen AVISPA ont publié leurs travaux de développement d'une plateforme contenant quatre outils d'analyse de protocoles et permettant la détection des attaques logiques sur les protocoles de sécurité. Cette plateforme suggère aussi des améliorations assurant la validité des propriétés de confidentialité et d'authentification [59] La structure de l'outil AVISPA est représenté sur la Figure suivante :

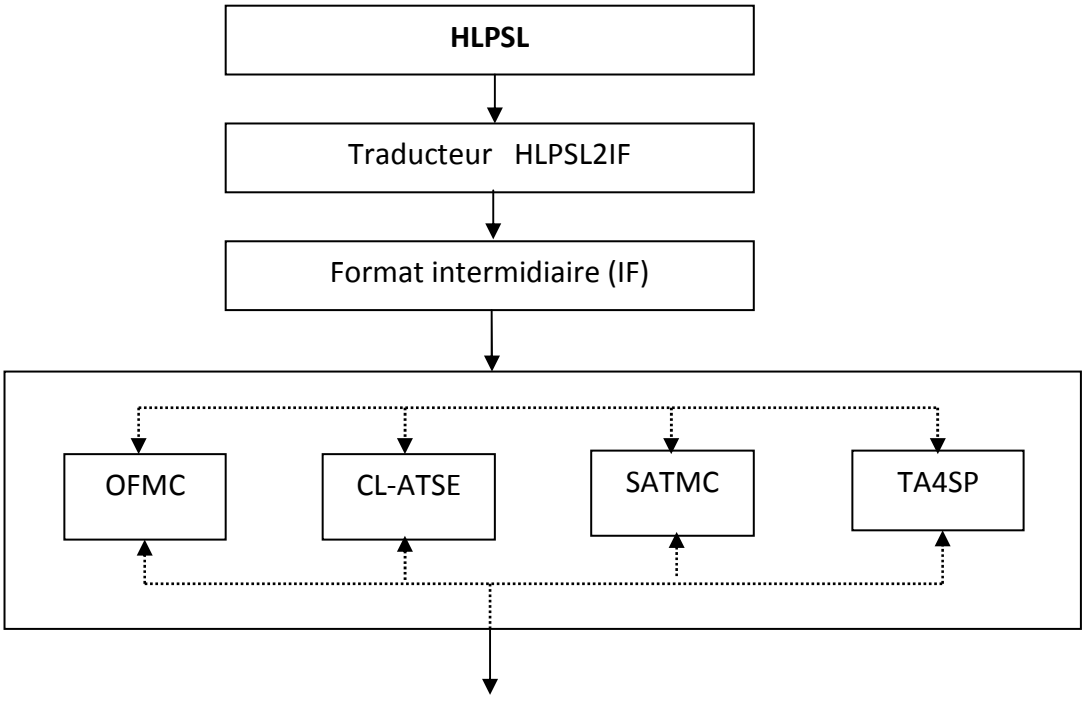


Figure IV.1 : La structure de l'outil AVISPA

L'outil supporte quatre moteurs de vérifications dis "backends" :

The On-the-fly Model-Checker (OFMC): Utilise plusieurs techniques symboliques pour explorer l'espace d'états à la demande.

CL-AtSe (Constraint-Logic-based Attack Searcher): Applique la résolution, de contraintes avec heuristiques de simplification et techniques d'élimination de redondances

The SAT-based Model-Checker (SATMC): Construit une formule propositionnelle qui encode toutes les attaques possibles (de longueur limitée) sur le protocole et soumet le résultat à un solveur

SAT TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols): Estime les connaissances de l'intrus en utilisant un langage d'arbre régulier avec réécriture pour produire des sous et sur-approximations

Une spécification HLPSL est traduite au format intermédiaire (IF), en utilisant un traducteur appelé hlpsl2if. Notez que cette étape de traduction intermédiaire est transparente pour l'utilisateur

Environnement graphique

AVISPA offre également un environnement graphique de mise au point des protocoles comme illustré sur la figure suivante :

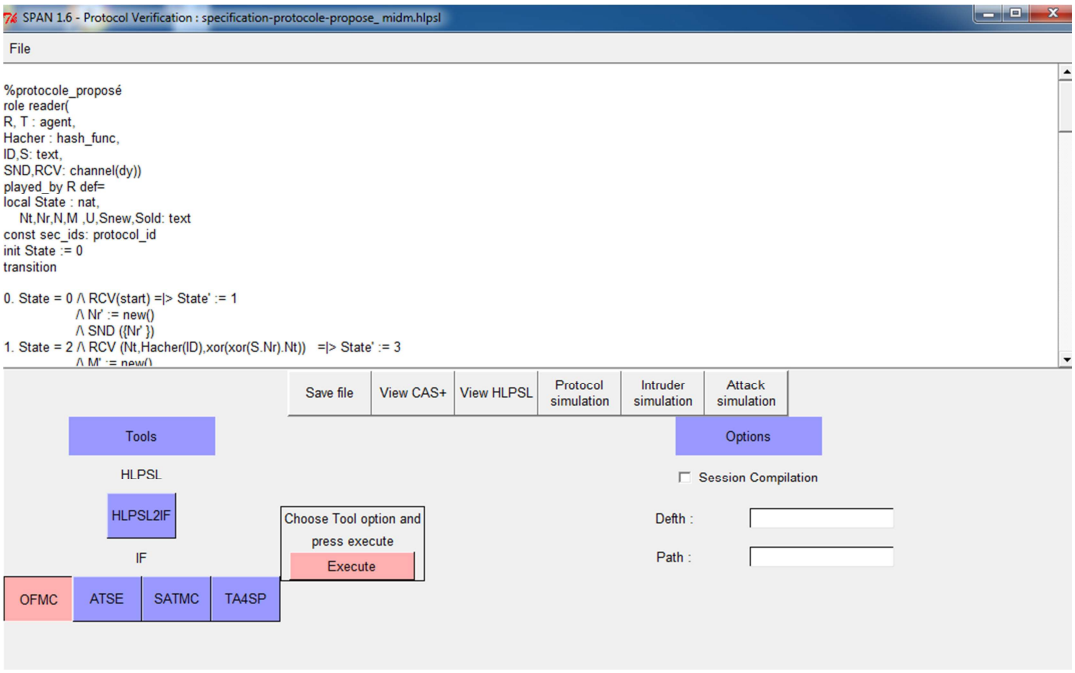


Figure IV.2 : Environnement Graphique d'AVISPA

Le langage HLPSL :

HLPSL (High Level Protocol Specification Language) est un langage formel de spécification modulaire du protocole de haut niveau basé sur des descriptions de rôles. Il supporte des primitives cryptographiques différentes (Clés symétriques et asymétriques, les fonctions de hachage) et de leurs propriétés algébriques (ou exclusif, exposant).

Le but de ces spécifications étant de pouvoir vérifier des propriétés de sécurités, l'authentification et la confidentialité. L'idée principale est de représenter un protocole cryptographique par un système d'états/transitions pour lequel il est possible de vérifier des propriétés de sécurité exprimées en logique temporelle linéaire (LTL).
Les transitions définissent le comportement du protocole de sécurité, et ainsi, à partir de l'état initial, nous sommes capables d'énumérer les états atteignables du protocole étudié.

Les spécifications HLPSL de protocoles sont divisées en rôles, ces derniers sont répartis en deux catégories distinctes :

Les rôles basiques : représente les agents participants aux protocoles
Les rôles composition : représente les scénarios des rôles basiques.

A la fin de la spécification, on détermine les propriétés de sécurité à vérifier.

Structure d'une spécification HLPSL

Une spécification HLPSL d'une communication entre deux agents Alice & Bob est structurée comme suite (à noter en HLPSL le symbole % signifie début de commentaire): On commence toujours par la spécification des rôles basiques, ici rôle Alice et rôle Bob[59]

role Alice (arguments)

played_by alice

----- }
----- } Liste des déclarations
----- }
----- }

Transition

----- }
----- } Les transitions de ''Alice''
----- }

end role

role Bob (arguments)

played_by Bob

----- }

----- Liste des déclarations

Transition

----- } Les transitions de ''Bob''
-----

end role

% on passe maintenant à la spécification des rôles de composition: rôle session et rôle %envirenment

role session (arguments)

----- } Listes des déclarations
-----

composition

Alice(les arguments) /\ Bob (les arguments) } composition de la session de communication entre Alice et Bob

end role

role environment( )

----- } Liste des déclarations et les connaissances initiales de l'intrus
-----

Composition

session(arguments) /\ session(argument) ----- } Établissement des sessions de communication entre Alice, Bob et intrus (spécification d'attaque)

end role

% On passe maintenant à la déclaration des propriétés à vérifier: goal

Goal

} Les propriétés à vérifier

end goal

environment()

Caractéristiques de HLPSL

Le langage HLPSL possède plusieurs caractéristiques, citons:

- support cryptographiques variés (clés symétriques, clés publiques, fonctions de hachage).
- information typée (ou non), avec des types simples ou composés.
- propriétés algébriques supportées (concaténation, OU exclusif, exponentiation).
- canaux pour les échanges de messages.

Comme on remarque, le langage de spécification HLPSL ne supporte pas quelques caractéristiques importantes comme:

- Les opérateurs arithmétiques: +, -, *, / …
- L'opérateur logique: ou (\/)
- Les fonctions de décalages, rotations, permutation ….
- Les boucles : if…else, while, repeat
- Le choix: Case.

IV.5.La vérification formelle du protocole proposé :

Dans cette section, on vérifie les propriétés de la confidentialité de l'identificateur ID et Le S (sec_ID et sec_S respectivement), l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader) du protocole propose décrit dans le chapitre III.

Ces propriétés de securite a verifier sont spécifiées dans HLPSL au niveau de la partie goal comme suit:

```
goal
      secrecy_of sec_id, sec_s
      authentication_on aut_tag
      authentication_on aut_reader
end goal
```

la spécification des rôles basiques :

```
%protocole_proposé
role reader(
R, T : agent,
Hacher : hash_func,
PRNG:function,
ID,S: text,
SND,RCV: channel(dy))
played_by R def=
local State : nat,
  V,H, Nt,Nr,N,M ,U,Snew: text
const sec_s: protocol_id
init State := 0
transition

0. State = 0 /\ RCV(start) =|> State' := 1
              /\ Nr' := new()
              /\ SND (Nr' )
             /\ secret(Nr,sec_s,{R,T})


1. State = 2 /\ RCV (Nt,H,V)   =|> State' := 3
              /\ V':= PRNG(xor(xor(S.Nr).Nt))


2. State = 2 /\ equal (V,V')   =|> State' := 5
              /\ M' := PRNG(Snew.Nr)
              /\ N' := PRNG(M)
              /\ SND (N)


3. State = 4 /\ RCV (start) =|>  State' := 7
              /\ U' := Hacher( Snew.M )
              /\ Snew' := xor(Snew,U)
```

```
end role


role tag ( T,R: agent,
Hacher : hash_func,
PRNG:function,
ID,S:text,
SND,RCV: channel(dy))
played_by T def=
local State : nat ,
    Nt,Nr,N,M ,U,Snew,V,H: text
const sec_id: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(Nr) =|> State' := 1
                /\ Nt' := new ()
                /\ V' := PRNG(xor(xor(S.Nr).Nt))
                /\ H' := Hacher(ID)


2. State = 2 /\ RCV (start) =|> State' := 3
                /\ SND(H,V,Nt)
                /\ RCV(N)
                /\ secret(Nt,sec_id,{T,R})


3. State = 4 /\ RCV (start) =|> State' := 5
                /\ M' := PRNG(S.Nr)
                /\ N' := PRNG(M)


4. State = 6 /\ equal(N,N') =|> State' := 7
                /\ U' := Hacher( S.M )
                /\ S' := xor(S , U)
end role
```

la spécification des rôles de composition: rôle session et rôle environment

```
role session(T,R : agent,Hacher: hash_func ,PRNG:function,ID,S:text) def=
local St,Rt,Sr,Rr : channel(dy)

composition
tag(T,R,Hacher,PRNG,ID,S,St,Rt) /\ reader(R,T,Hacher,PRNG,ID,S,Sr,Rr)
end role
```

Concernant l'authentification, il y a deux attaques possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle. Cela est spécifié dans le rôle environnement dans HLPSL comme suit :

Ataque Main-in-the-middle :

Le scenario spécifié dans le rôle environnement ci-dessous permet de détecter des attaques du type Main-in-the-midlle s'il existe :

```
role environment() def=
const t,r : agent,
hach: hash_func,
prng:function,
id,idi,si,s: text,

aut_tag, aut_reader:protocol_id
intruder_knowledge = {t,r,hach,prng,idi,si}
composition

%attaque MITM
session(t,r,hach,prng,id,s)   /\  session(t,i,hach,prng,idi,si) /\  session(i,r,hach,prng,idi,si)
end role

environment()
```

Après la vérification de ce protocole par l'outil CL-AtSe d' AVISPA, le résultat est comme Suit **:**

```
% Version of 2006/02/13

SUMMARY

  SAFE

DETAILS

  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL

  C:\progra~1\SPAN\testsuite\results\protocole propose ( attaque MITM).if

GOAL

  as_specified

BACKEND

  OFMC

COMMENTS

STATISTICS

  parseTime: 0.00s
```

searchTime: 0.03s

visitedNodes: 4 nodes

depth: 2 plies

Ce résultat signifie qu'il n y a pas d'attaque Main-in-the-Middle.

Attaque par rejeu :

Dans l'attaque par rejeu (Replay Attack), l'adversaire peut écouter le message de réponse du tag et du lecteur. Il retransmettra le message écouté sans modification au lecteur plus tard. La spécification ci-dessous du rôle environnement en HLPSL dépend du traitement de deux sessions en parallèles (représenter par le symbole /\ ) entre deux tags légitimes et un même lecteur (t1, t2 et r). Ce scenario permet de détecter les attaques du type "Attaque par rejeu" s'elles existent.

```
role environment() def=
const t1,t2,r : agent,
hach: hash_func,
prng:function,
idt1,idst1,idt2,idst2: text,

aut_tag, aut_reader:protocol_id
intruder_knowledge = {t1,t2,r,hach,prng}
composition

%attaque par rejeu
session(t1,r,hach,prng,idt1,idst1)   /\  session(t2,r,hach,prng,idt2,idst2)

end role

environment()
```

Après la vérification de ce protocole ,le résultat est comme suit:

```
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\specification-protocole-propose_ midm.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 4 nodes
```

depth: 2 plies

Ce résultat signifie qu'il n y a pas d'attaque par rejeu.

IV.6 La vérification formelle du protocole EKE :

Dans cette section, on vérifie les propriétés de la confidentialité, l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader) du protocole EKE décrit dans le chapitre III.
Ces propriétés sont spécifiées dans HLPSL au niveau de la partie goal comme suit:

```
end role

goal

 secrecy_of sec_k1, sec_k2
 authentication_on nb
 authentication_on na

end goal
```

la spécification des rôles basiques :

```
role alice (A,B: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by A
def=

  local State   : nat,
      Ea,Na,Nb,K: text

 const sec_k1 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\ Rcv(start) =|>   State' := 1
              /\ Ea' := new()
              /\ Snd({Ea'}_Kab)

   2. State = 1  /\ Rcv({{K'}_Ea}_Kab) =|>  State' := 2
               /\ Na' := new()
               /\ Snd({Na'}_K')
              /\ secret(K',sec_k1,{A,B})
               /\ witness(A,B,na,Na')

   3. State = 2 /\ Rcv({Na.Nb'}_K)  =|>  State' := 3
              /\ Snd({Nb'}_K)
              /\ request(A,B,nb,Nb')
```

```
end role


role bob (B,A: agent,
          Kab: symmetric_key,
          Snd,Rcv: channel(dy))
played_by B
def=

  local State   : nat,
      Ea,Na,Nb ,K: text

const sec_k2 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\ Rcv({Ea}_Kab) =|> State' := 1
            /\ K' := new()
            /\ Snd({{K'}_Ea}_Kab)
           /\ secret(K',sec_k2,{A,B})

   2. State = 1 /\ Rcv({Na}_K) =|> State' := 2
            /\ Nb' := new()
            /\ Snd({Na'.Nb}_K)
            /\ witness(B,A,nb,Nb)

   3. State = 2 /\ Rcv({Nb}_K)=|>  State' := 3
            /\ request(B,A,na,Na)

end role
```

la spécification des rôles de composition: role session et role envirenment :

```
role session(A,B: agent,
          Kab: symmetric_key)
def=

  local SA, RA, SB, RB: channel (dy)

  composition
    alice(A,B,Kab,SA,RA) /\ bob(B,A,Kab,SB,RB)

end role
```

Concernant l'authentification, il y a deux attaques possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle. Cela est spécifié dans le rôle environnement dans HLPSL comme suit :

Attaque par rejeu :

. La spécification ci-dessous du rôle environnement en HLPSL permet de détecter les attaques du type "Attaque par rejeu" si elles existent.

```
role environment()
def=

  const a, b ,r   : agent,
        kab    : symmetric_key,
        na, nb : protocol_id,
kt11,kt21:symmetric_key

  intruder_knowledge={a,r}

  composition

%attaque par rejeu

session(a,r,kt11)   /\  session(b,r,kt21)

end role
```

Après la vérification de ce protocole par l'outil OFMC d' AVISPA, le résultat est comme :

```
suit:
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\EKE par rejeu.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.15s
  visitedNodes: 49 nodes
  depth: 12 plies
```

Ce résultat signifie qu'il n y a pas d'attaque par rejeu.

Attaque Main-in-the-middle :

Le scenario spécifié dans le rôle environnement ci-dessous permet de détecter des attaques du type Main-in-the-midlle s'il existe :

```
role environment()
def=

  const a ,r   : agent,
       kab    : symmetric_key,
       na, nb : protocol_id,
k:symmetric_key
intruder_knowledge={a,r}

  composition

%attaque MITM

session(a,r,k)   /\  session(a,i,k) /\  session(i,r,k)

end role
```

Après la vérification de ce protocole par l'outil OFMC d' AVISPA, le résultat est comme suit:

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\EKE MIDM avec attaque.if
GOAL
  authentication_on_nb
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.01s
  searchTime: 0.07s
  visitedNodes: 29 nodes
  depth: 4 plies
```

Ce résultat signifie que le Protocol EKE est vulnérable à l'attaque man in the middle ,et l'attaque du lecteur (authentication_on_nb)

IV.7 Proposition d'une amélioration du protocole EKE :

Comme nous l'avons vérifié formellement, le protocole EKE ne résiste pas a l'attaque d'authentification main in-the-middle .

Dans cette section, on va présenter une amélioration du protocole EKE. Pour laquelle on vérifie la confidentialité, l'authentification du tag et l'authentification du lecteur par les outils AVISPA et SPAN. Le protocole conçu résiste a l'attaque main in-the-middle.
En comparant l'ordre des messages transmis dans le protocole EKE avec celui du protocole proposé et vérifie auparavant on remarque que authentification du tag auprès du lecteur est faite au départ comme suit :

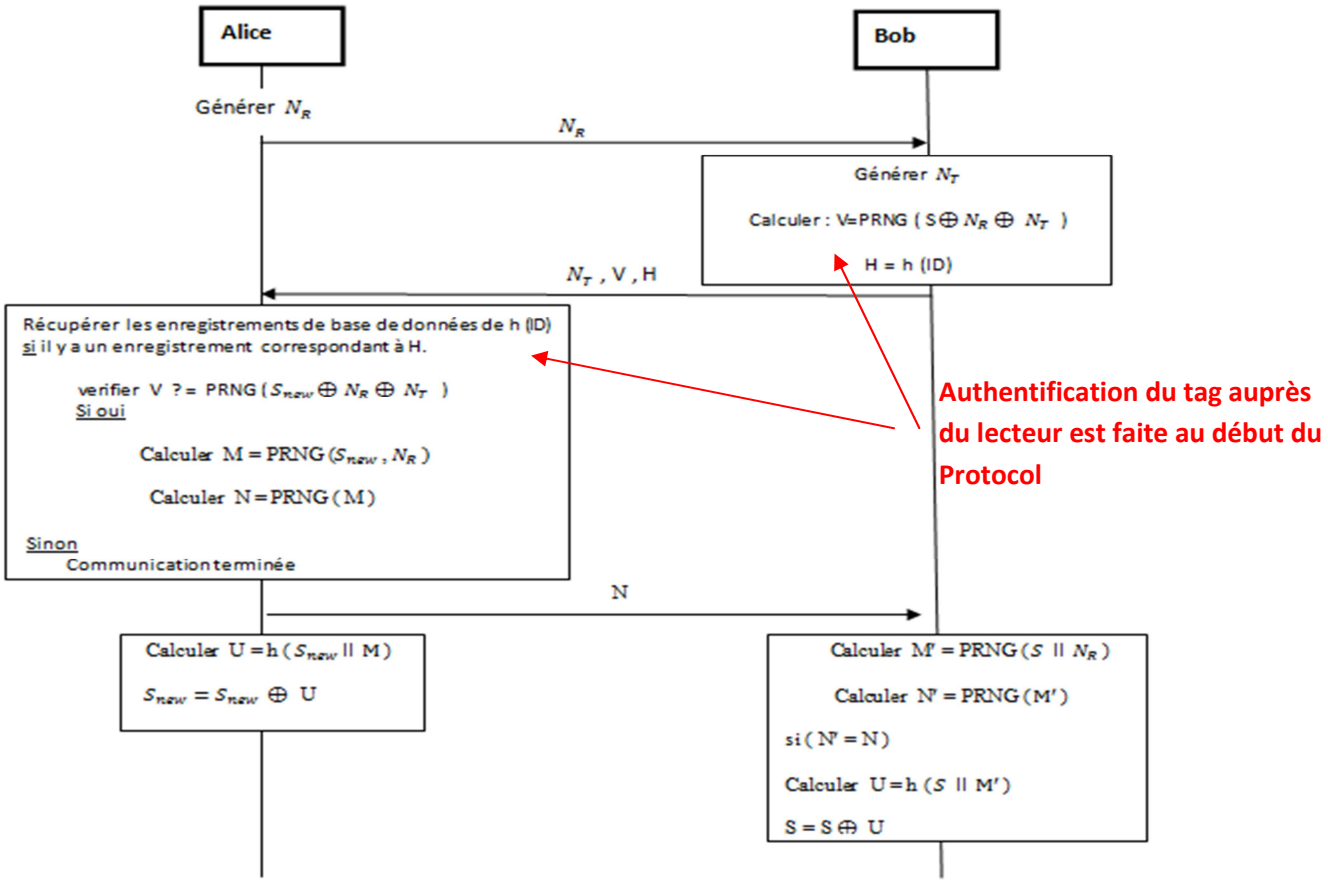


Figure IV.3 : Schéma générale de protocole proposé

Tandis que dans le protocole EKE l'authentification du tag est faite a la fin du protocole dans l'étape 5

Etape 5 :

Le lecteur décrypte le message pour obtenir le Na et le compare au Na déjà généré : si ils correspondent donc l'étiquette est authentifié
Puis envoie { Nb }_K a l'étiquette :

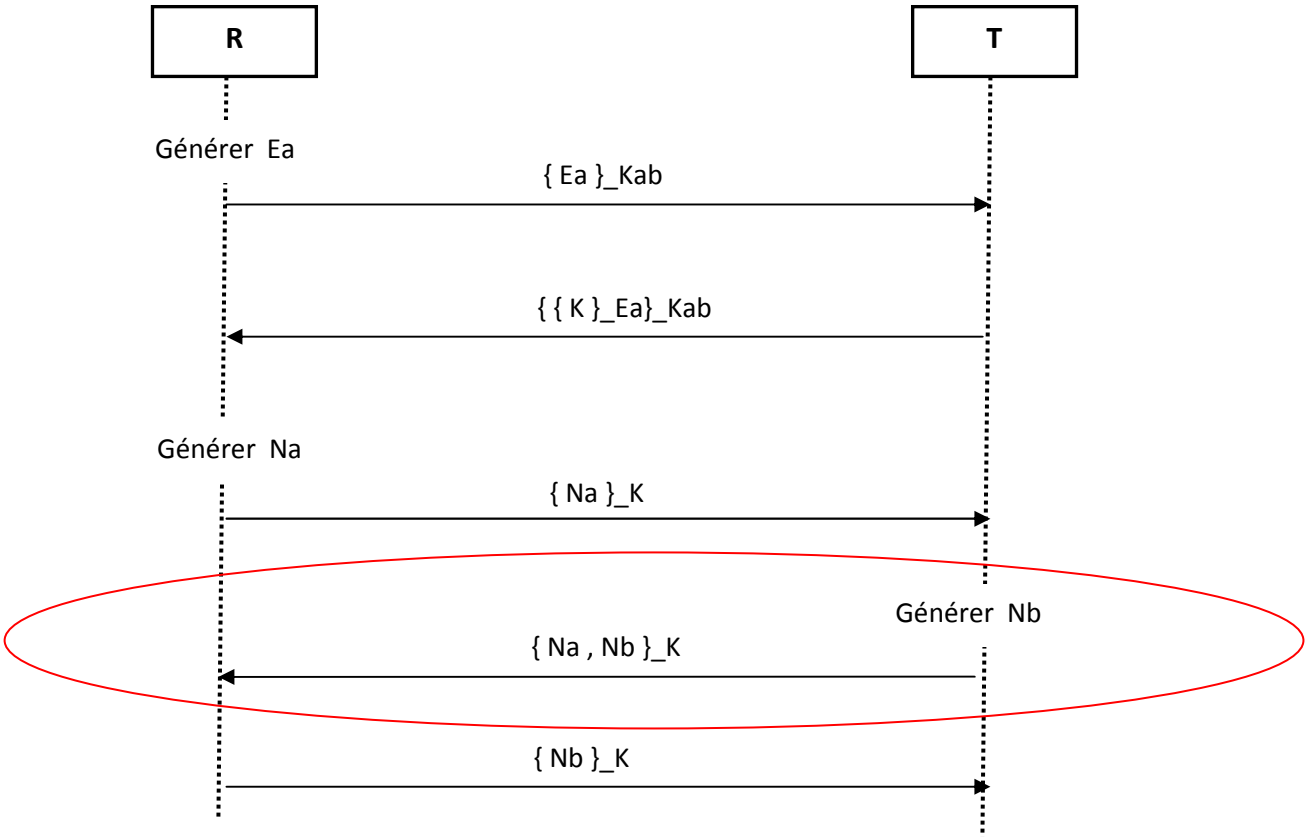


Figure IV.4: le protocole EKE

Pour le protocole EKE on propose l'authentification du tag auprès du lecteur juste au début de la communication : le lecteur reçoit l'identificateur du tag et le compare a celui déjà enregistre dans la base de données :

- s'il y a un enregistrement correspondant à Id-tag alors le tag est authentifié.
- Sinon communication sera terminée.

Comme le montre le schéma suivant :

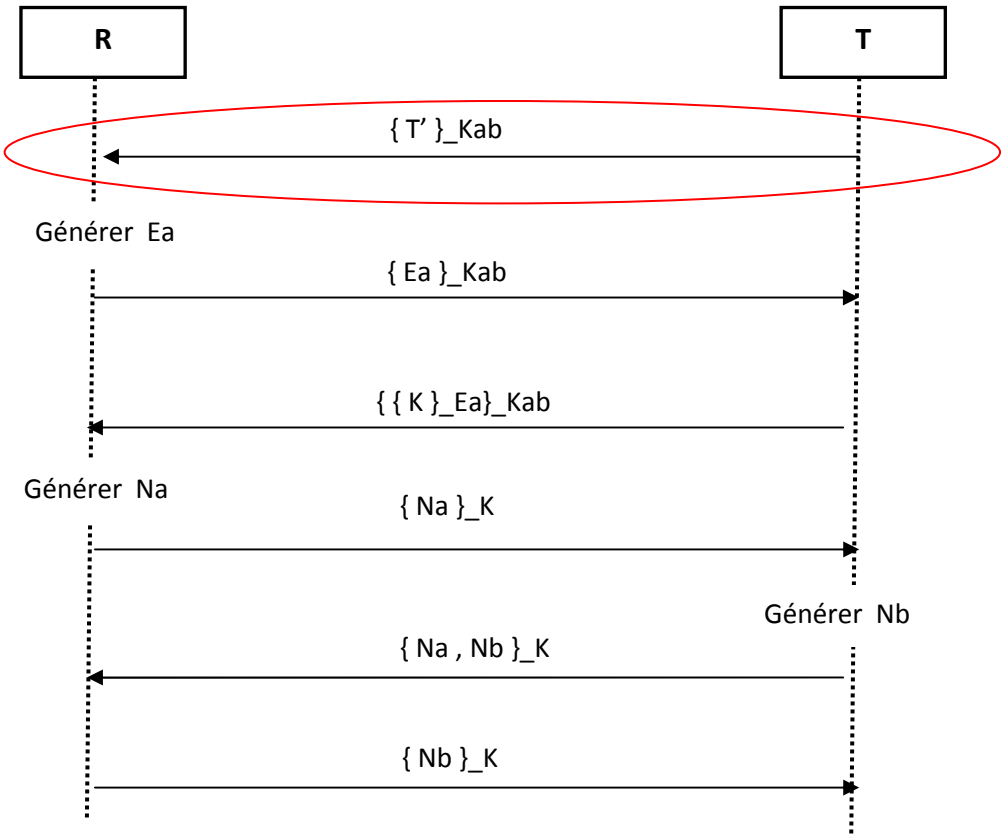
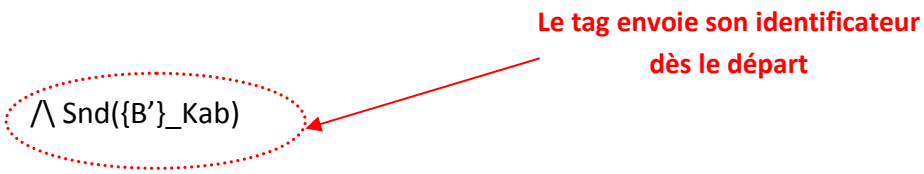


Figure IV.5: le protocole EKE amélioré

La partie modifiée dans la spécification HLPSL est les rôles de base :

```
role alice (A,B: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by A
def=

  local State   : nat,
      Ea, Na,Nb,K: text

 const sec_k1 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\  Rcv(start)  /\  Rcv({B'}_Kab)  =|>   State' := 1
                  /\ Ea' := new()
                  /\ Snd({Ea'}_Kab)

   2. State = 1  /\ Rcv({{K'}_Ea}_Kab) =|>  State' := 2
              /\ Na' := new()
              /\ Snd({Na'}_K')
              /\ secret(K',sec_k1,{A,B})
              /\ witness(A,B,na,Na')

   3. State = 2 /\ Rcv({Na.Nb'}_K)  =|>  State' := 3
             /\ Snd({Nb'}_K)
             /\ request(A,B,nb,Nb')

end role


role bob (B,A: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by B
def=

  local State   : nat,
       Ea,Na,Nb ,K: text

 const sec_k2 : protocol_id

  init  State := 0

  transition

   1.   State = 0 /\ Rcv({Ea'}_Kab) =|> State' := 1
```

Le lecteur reçoit l'identificateur du tag dés le départ



/\ Snd({B'}_Kab)

/\ K' := new()
/\ Snd({{K'}_Ea'}_Kab)
/\ secret(K',sec_k2,{A,B})

2. State = 1 /\ Rcv({Na'}_K) =|> State' := 2
/\ Nb' := new()
/\ Snd({Na'.Nb'}_K)
/\ witness(B,A,nb,Nb')

3. State = 2 /\ Rcv({Nb}_K)=|> State' := 3
/\ request(B,A,na,Na)

end role

Après la vérification de protocole EKE amélioré par l'outil OFMC d' AVISPA, le résultat est comme suit:

```
% OFMC

% Version of 2006/02/13

SUMMARY

  SAFE

DETAILS

  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL

  C:\progra~1\SPAN\testsuite\results\EKE MIDM corriger.if

GOAL

  as_specified

BACKEND

  OFMC

COMMENTS

STATISTICS

  parseTime: 0.00s

  searchTime: 0.01s

  visitedNodes: 1 nodes
```

IV.7 Comparaison de la sécurité :

Le Tableau suivant illustre une comparaison de la sécurité avec les protocoles vérifiés :

protocole	EKE	EKE amélioré	Le protocole proposé
Résistance à l'attaque par rejeu	oui	oui	oui
Résistance à l'attaque MITM	non	oui	oui

Tableau IV.1: Analyse de la sécurité.

Complexité du tag :

La complexité de toutes les implémentations de primitives cryptographiques exigées devrait être la plus faible possible , d'où le coût du tag, i.e. complexité du tag, sera aussi faible. Le tableau suivant illustre les primitives cryptographiques exigées dans le tag selon le protocole d'authentification :

Protocole	**EKE**	**Le protocole proposé**
Fonction de hachage		x
PRNG	x	x
Cryptage symétrique	x	

Tableau IV.2: complexité du tag.

Dans notre travail, le protocole d'authentification EKE exige le chiffrement symétrique, généralement les algorithmes symétrique implémentés dans les systèmes RFID sont les algorithmes de catégorie chiffrement par bloque (block cipher) comme algorithme AES ,sa mise en œuvre a été réalisée en utilisant environ 3400 portes logiques sous forme de blocs de taille 128 bits (avec une fréquence d'horloge maximale estimée à 80MHz et la consommation d'énergie 8.2 µA dans 100kHz).

Le protocole proposé exige une fonction de hachage qui est une primitive cryptographique. Yüksel [13] a présenté l'implémentation de faible coût des fonctions de hachage, en utilisant seulement 1700 portes logiques sous forme de blocs de taille 64 bits (avec une fréquence d'horloge maximale estimée à 100 MHz).

Les deux protocoles étudiés auparavant exigent un générateur des nombres pseudo-aléatoire (PRNG) qui sert à générer des nonces. La mise en œuvre de ce générateur peut appeler une fonction de hachage .

Donc, en ce qui concerne la complexité, le tag du protocole proposé est de coût bas par rapport au tag du protocole EKE.

IV.8 Conclusion :

La vérification formelle est une méthode qui permet la description mathématique d'un protocole pour afin de vérifier la fiabilité des protocoles cryptographiques.

Dans ce chapitre nous avons présenté quelques notions de base de la vérification ainsi que ses objectifs, puis on a présente la vérification formelle : ses caractéristiques et ses méthodes ainsi, un ensemble d'outils qui permet d'assurer cette vérification, nous avons basé sur l'outil AVISPA, et le langage HLPSL .
Ensuite nous avons vérifié formellement le protocole cryptographique propose en utilisant les outils AVISPA & SPAN. Nous avons montré l'importance de cette vérification pour assurer les propriétés de confidentialité et d'authentification dans les systèmes RFID. Suite à nos vérifications, nous avons constaté que notre protocole est sûr.

Références bibliographiques

[1]: Nicolas Seriot. << Les systèmes d'identification radio (RFID) fonctionnement, applications et dangers >>. Article, IL-2005B, 13 janvier 2005.
[2]: Klaus finkenzeller. << RFIF handbook fundamentals and applications in contactless smart cards, radio frequency identification and near field communication >>. Book, third edition page 2-7, 2010.

3]: Jeremy Landt. << The history of rfid. Potentials >>, IEEE, 24(4):8-11, Oct 2005.

[6] : Belrepayre Sylvain. << La technologie RFID et le protocole Modbus >>. Exposé, université PARIS-EST, 01/02/2013

[7]: BRUN-MUROL Pierre. << Vers une méthodologie normalisée d'évaluation des solutions RFID en application de sécurité >>. Mémoire présenté en vue de l'obtention du diplôme de maitrise sciences appliquées (génie informatique), Ecole polytechnique de Montréal, avril 2013.
[8]: Harvey Lehpamer. << RFID Design Principles >>. ARTECH HOUSE, Boston, 2008.

12]: BACHOTI Youssef, BELHAJ SENDAGUE Bassim, RODRIGUES OLIVEIRA et Joao Gabriel. <<PROJET RFID >>**.** Projet de fin d'étude. Paris Tech : Institue des sciences et technologies, 25 janvier 2011.

[13]: Stevan Preradovic and Nemai Chandra Karmakar. <<Rfid transponders>>. In International Conference on Electrical and Computer Engineering, pages 96–99, Dec 2006.

[14] : Samuel Fosso Wamba. << Les impacts de la technologie RFID et du réseau EPC sur la gestion de la chaine d'approvisionnement >>. Thèse de doctorat, université de Montréal, septembre 2009.

[15]: Dat Son Nguyen. << Développement des capteurs sans fil basés sur les tags RFID uhf passifs pour la détection de la qualité des aliments >>. Thèse de doctorat, Soutenue a Université de Grenoble, Septembre 2013.

[16]: Pierre-Henri THEVENON. << Sécurisation de la couche physique des communications sans contact de type RFID et NFC>>. Thèse de doctorat, université de Grenoble, 10 novembre 2011.

[17] : Anthoy Ghiotto. << Conception d'antennes de tags RFID UHF, Application a la réalisation par jet de la matière >>. Thèse de doctorat, université de Grenoble, novembre 2008.

[18]: Sanjay Sarma. << A history of the EPC in RFID: Application, Security and Privacy >>. Chapitre extrait de: RFID: applications, security, and privacy, pages 37–55. Addison-Wesley, Boston, London, Dec 2005.

[19]: Dat Son Nguyen. << Développement des capteurs sans fil basés sur les tags RFID uhf passifs pour la détection de la qualité des aliments >>. Thèse de doctorat, Soutenue a Université de Grenoble, Septembre 2013.

[20]: Mandeep Kaur, Manjeet Sandhu, Neeraj Mohan and Parvinder S. Sandhu. << Rfid technology principles, advantages, limitations and its Applications >>. International Journal of

[21]: Marie Lise Flottes, LIRMM, Giorgio Di Natale, LIRMM, Guy Gogniat. << Sécurité Des Systèmes Embarqué >>. Article, Lab-STICC, 2011.

[22]: HAMADOU Sardaouna**.** << Analyse formelle des protocoles cryptographiques et flux d'information admissible >>. Thèse de doctorat, université de Montréal École Polytechnique, mars 2008.

[23]: Bernard COUSIN, << Sécurité des réseaux informatiques>>. Support de cours, université de Rennes 1.

[24]: Philippe MARTIN et Refik MOLVA. << Analyse de la sécurité >>. La source de la section : Analyse de la sécurité du document SP 1.2. (Étude prospective des besoins dans un réseau RFID communautaire) du projet PAC-ID GD, http://www.eurecom.fr/~martinph/PACID/SP1_2secur.html.

[25]: Jérémy Briffaut. << Formalisation et garantie de propriétés de sécurité système : application à la détection d'intrusions >>. Thèse de doctorat, universite d'Orleans déparement informatique, 13 décembre 2007.

[26]: Mathieu Baudet. << Sécurité des protocoles cryptographiques : aspects logiques et calculatoires >>**.** Thèse de doctorat, université de Cachan, 16 janvier 2007.

[27]: Heinrich Hordegen. << Vérification des protocoles cryptographiques : Comparaison des modèles symboliques avec une application des résultats — Etude des protocoles récursifs >>. Thèse de doctorat, université de Nancy, 29 Novembre 2007.

[29]: https://ssi.ac-strasbourg.fr/

[30]: G.Avoine and P.Oechslin. <<RFID traceability: A multilayer problem>>. In A.Patrick and M.Yung, editors, Financial Cryptography -FC'05, pages 125-140. Springer-Verlag, 2005.

[31]: A.Juels, S.Garfinkel, and R.Pappu. <<RFID privacy: An overview of problems and proposed solutions>>. IEEE Security and Privacy, 3(3):34-43, May/June 2005.

[32]: A.Karygiannis, T.Phillips, A.Tsibertzopoulos. <<RFID security: A taxonomy of risk>>. In Proceedings of the 1st international conference on communications and networking in China (ChinaCom'06) (pages 1-7). Beijing: IEEE. 2006.

[33]: Ding Zhen-hua, Li Jin-tao, and Feng Bo. << A Taxonomy Model of RFID Security Threats>>. Communication Technology, ICCT, pages 765-768, November 2008.

[34]: L. Mirowski, J. Hartnett, and R. Williams. <<An RFID Attacker Behavior Taxonomy>>. IEEE Pervasive Computing, 8(4):79-84, pages 79-84, October/December. 2009.

[35]: A.Mitrokotsa,MR. Rieback,AS. Tanenbaum. <<Classifying RFID attacks and Defenses>>. Special Issue on Advances in RFID Technology, Information Systems Frontiers, Springer Science and Business Media, LLC 2009. doi: 10.1007/s10796-009-9210-z., July 2009.

[36]: A. Mitrokotsa, M. Beye, P. Peris-Lopez, chapter: <<Security Primitive Classification of RFID Attacks>>. In Book: Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Eds. D. Ranasinghe, M. Sheng, S. Zeadally. Springer-Verlag. 2011.

[37]: Hong Li, YongHui Chen, ZhangQing He. <<The Survey of RFID Attacks and Defenses>>. Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, vol., no., pages 1-4, 21-23, September 2012.

[38]: Torstein HAVER. <<Security and Privacy in RFID Applications>>. Thèse de Master, Norwegian University of Science and Technology Department of Telematics, juin 2006.

[39]: Coulton, P., Rashid, O., and Bamford, W. << Experiencing 'touch' in mobile mixed reality games, Proceedings of The Fourth Annual International Conference in Computer Game Design and Technology>>, Liverpool, Novembre 2006.

[40]: Tuyls, P. & Batina, L. <<RFID-tags for anti-counterfeiting>>, In D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006, Vol. 3860 of LNCS, pp.115-131, Springer-Verlag, ISSN: 0302-9743, San Jose, CA, USA.

[41]: Mojtaba Alizadeh, Mazdak Zamani, Ali Rafiei Shahemabadi,JafarShayan and Ahmad Azarnik. <<A Survey on Attacks in RFID Networks>>. Open International Journal of Informatics (OIJI), VOL1, Universiti of Malaysia, 2012

[43]: Monty L. << Vulnérabilités des RFID >>, Article, 2010.

[44]: Stefan Brands and David Chaum. <<Distance-bounding protocols>>. Advances in Cryptology EUROCRYPT '93, Springer-Verlag LNCS 765, pp 344–359, May 1993.

[45]: Léonard Gross. << Sécurité RFID et préservation de la sphère privée >>. Mémoire de fin d'étude, Institute d'Information et de Communication, Suisse, Janvier 2007.

[**46]**: Rieback, M. R.; Crispo, B. & Tanenbaum, A. S. <<Is your cat infected with a computer virus? >>. Proc. of the 4th Annual IEEE International Conference on Pervasive Computing and Communication, pp. 169-179, ISBN: 0-7695-2518-0, 13-17 March 2006, Pisa, Italy.

[47]: Xavier Lemarteleur. <<Traçabilité contre vie privée : les RFIDs Ou l'immixtion des technologies dans la sphère personnelle>>. Mémoire de fin d'étude, université Paris II – Panthéon / Assas, octobre 2004.

[48]: Dirik Henrici. <<RFID security and privacy, concepts, protocols, and architectures>>. pages 45-55, 2008.

[49]: Mohammad Reza Sohizadeh Abyaneh. << security analysis of lightweight schemes for RF ID systems>>. Thèse de doctorat, université de Bergen norway, juin 2012.

[50]: Xu Zhuang · Yan Zhu & Chin-Chen Chang. << A New Ultralightweight RFID Protocol for Low-Cost Tags: R2AP>>. Wireless Pers Commun (2014) 79:1787–1802 DOI

10.1007/s11277-014-1958-x. Springer Science+Business Media New York 2014, 26 July 2014.

[51]: Ha, J., S.-J. Moon, J. M. G. Nieto and C. Boyd, <<Low-cost and strong-security RFID authentication protocol>>, in: EUC Workshops, 2007.

[52]: T. van Deursen and S. Radomirović. << Attacks on RFID Protocols>>. In Proc. 4th International Workshop on Security and Trust Management (STM'08), ENTCS. Elsevier, August 2009.

[53]: T. van Deursen and S. Radomirović. <<Security of RFID protocols- A case of study>>. In Proc. 4th International Workshop on Security and Trust Management (STM'08), ENTCS. Elsevier, Juin 2008.

[54]: Hung-Yu Chien and Chen-Wei Huang. <<A lightweight RFID protocol using substring>>. In Embedded and Ubiquitous Computing (EUC), 2007.

[55]: Lee et al. <<RFID mutual authentication scheme based on synchronized secret Information>>. In Symposium on Cryptography and Information Security, Hiroshima, Japan, January 2006.

[56]: Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. <<LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags>>. In *Proceedings of 2nd workshop RFID security*. Graz, Austria: Ecrypt. 2006.

[57] : Pascal Lafourcade. << Vérification de protocoles cryptographiques en présence de théories équationnelles>>. Thèse de doctorat, école normale supérieure de CACHAN Laboratoire Spécification et Vérification CNRS UMR 8643, lundi 25 septembre 2006.

[58]: Amina Cherif, Damien Sauveron and Malika Belkadi. << Overview on Formal Verification Methods for RFID Protocols >>. Article, université de Tizi Ouzou, Laboratoire LARI,Computer Science Department, Algeria, Université de Limoges, XLIM UMR CNRS 7252 – Mathematics and Computer Science Department, Limoges, France. 2016.

[59]: Hamdi NASSER, Michel FANGAYOUMANI et Simon DEBRAS. << Vérification formelle d'un protocole de sécurité à l'aide d'un outil d'analyse automatique des failles de sécurité >>. Rapport de projet, ENAC, 8juin 2011.
Computer Society.journal, 1997.

III .1. Introduction :

Les techniques RFID sont en proie à la sécurité et problèmes de confidentialité dus au canal de communication sans fil et insécurisé, afin d'aborder les questions de sécurité des systèmes RFID, différentes solutions ont été proposée. Parmi ces dernières, les protocoles cryptographiques qui représentent la clé de base de la sécurité RFID.

Dans ce chapitre, Dans un premier temps nous allons présenter quelque protocole d'authentification dans les systèmes RFID, Ensuite, nous proposons un protocole d'authentification qui utilise des générateurs de nombres pseudo-aléatoires (PRNG) et quelques opérations cryptographiques simples.

III.2. Les protocoles cryptographiques dans les systèmes RFID :

Les protocoles cryptographiques sont des petits programmes qui spécifient une séquence d'émissions/réceptions de messages qui visent à établir entre deux ou plusieurs participants (agents), des communications répondant à certaines propriétés de sécurité telles que la confidentialité, l'intégrité, l'authentification, ces protocoles utilisent des primitives cryptographiques à faible coût comme le bitwise opérations, générateurs de nombres pseudo-aléatoires, fonctions de hachage, etc .
Deux types de protocoles cryptographiques existant: légers (lightweight) et ultralégers (Ultralightweight) qui attirent beaucoup d'attention car ils sont plus adaptés pour les limites de ressources d'étiquettes RFID.

III.2.1 Les protocoles lightweight :

Ce type de Protocoles sont très utilises dans l'industrie du a leur avantage de maintenir la demande de calcul et le coût très faible des étiquettes RFID.La cryptographie lightweight est divisée en deux :

Primitives lightweight: cette catégorie regroupe les primitives symétriques, les primitives asymétriques, les fonctions de hachage et les générateurs de nombres aléatoires.

Protocoles lightweight: utilisent les primitives lightweight pour fournir des propriétés de sécurité. Cette catégorie peut être divisée en cinq sous- catégories: les protocoles d'identification, les protocoles d'authentification, protocoles de distance bounding, les protocoles de regroupement de preuve et protocoles de propriété du tag .

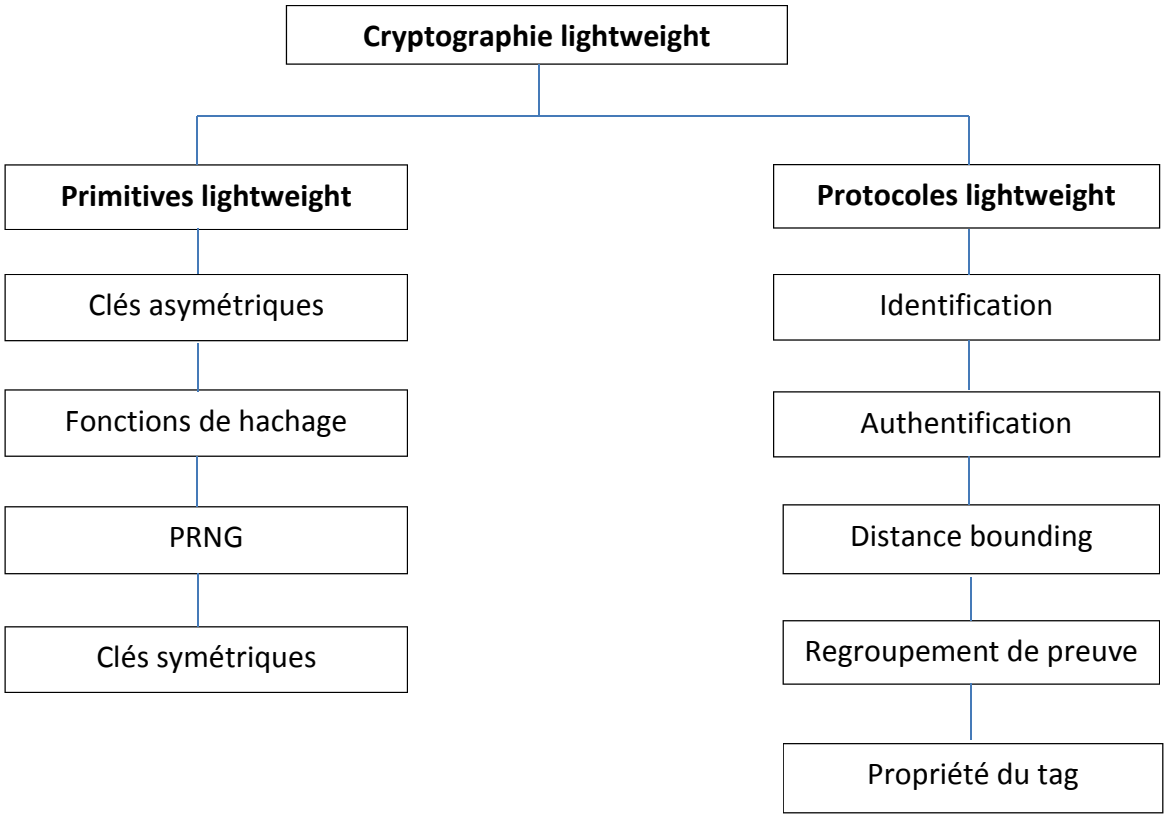


Figure III.1: Les protocoles légers dans les systèmes RFID.

III.2.2 Les protocoles ultralightweight

Ce sont des protocoles basés sur les opérations ultralégères visent à fournir une authentification sans utiliser des primitives cryptographiques et entraîner uniquement des opérations binaires simples et arithmétiques sur le tag (par exemple XOR, AND, OR, rotation, etc.).

Les protocoles ultralightweight sont divisés en deux groupes principaux: les protocoles basés sur "minimaliste cryptographie" et des protocoles basés sur des problèmes mathématiques NP-difficiles.III.3 Présentation de quelques protocoles d'authentification dans les systèmes RFID :

Au cours des dernières années, de nombreux protocoles d'authentification ont été proposes afin de surmonter les problèmes des systèmes RFID, on présente ici quelque uns :

III.3.1 Le protocole Fan et al :

Fan et Al ont propose un Protocol d'authentification mutuelle léger. dans ce protocole les trois composants du protocole pré-partagent le tuple (K_i, cro (·), Rot (·), PRNG(·))

Liste des notations utilisées :

La liste des notations utilisées dans le protocole fan et al est la suivante :

Notations	description
RID	Identifiant prive du tag
TID	Identifiant prive du lecteur
Nr , Nt, Ns	Numéro aléatoire généré par le lecteur l'étiquette et le serveur respectivement
Ki	La clé de la i-éme session
PRNG(.)	La fonction Pseudo Random Number Generator
Cro(x,y)	Le fonctionnement du bit cross
Rot(x,y	L'opération de rotation, x=W(y)
⊕	Opération OU exclusif.
Mark	Le statut de la dernière session.
||	Opération de concaténation.

Tableau III.1: liste des notations utilisées dans le protocole Fan et Al.

Tableau de données d'index :

La table de données d'index incluait la valeur d'index et l'index contenu qui sont uniques .Dans chaque session, la valeur de la clé est mise à jour, donc la valeur de l'index est fraîche pour chaque session. De plus, après chaque session réussie, le statut de Mark passe de "00" à "10".

Valeur d'index	Contenu d'index
cro(RID_TID ; K1)	Rot(K1_TID ; K1_RID)
cro(RID_TID ; K2)	Rot(K2_TID ; K2_RID)
: : :	: : :
cro(RID_TID ; Ki)	Rot(Ki_TID ; Ki_RID)
cro(RID_TID ; Ki+1)	Rot(Ki+1_TID ; Ki+1_RID)

Tableau III.2: table dindex du protocole Fan et Al.

Description du protocole :

Le protocole fan et al, comme indiqué sur la figure (), exécute les étapes suivantes :

tape1 :

Le lecteur démarre le protocole en envoyant un nombre aléatoire N_R à l'étiquette.

etape2 :

Après la réception de N_R , l'étiquette génère un nombre aléatoire N_t et place Mark = 00.

Elle transmet ensuite cro (RID $\oplus$ TID, K_i) , N_t au lecteur.

Etape3 :

Après avoir reçu le message, le lecteur obtient N_t et envoie cro (RID $\oplus$ TID, K_i) , N_t , N_R au serveur.

Etape4 :

Le serveur reçois N_R et N_t utilise cro (RID $\oplus$ TID, K_i pour trouver le contenu d'index correspondant dans l'IDT.

S'il peut trouver une correspondance, cela indique que la dernière session a été faite correctement et la session en cours est exécutable.

Il génère un nombre aléatoire N ss
envoie cro (RID $\oplus$ TID, $K_i \oplus N_S$) Rot ($K_i \oplus$ TID, $K_i \oplus$ RID) $N_S \oplus K_i$ au lecteur.

Sinon, le l'authentification échoue et le protocole sera terminé

Etape 5 :

Une fois que le lecteur a reçu le tuple (cro (RID $\oplus$ TID, $K_i \oplus N_S$) Rot ($K_i \oplus$ IDD, $K_i \oplus$ RID) $N_S \oplus K_i$), en fonction du poids W ($K_i \oplus$ TID) de l'opération de rotation et $K_i \oplus K_i \oplus$ TID il obtient TID.

Il obtient alors N_S et vérifie la valeur de cro (RID $\oplus$ TID, $K_i \oplus N_S$) en comparant avec la valeur reçue.

Si c'est le cas, il calcule TID $\oplus N_R$ et TID $\oplus N_S$ et les envoie à l'étiquette.

Etape 6 :

Après avoir reçu ce message, l'étiquette obtient N_S et

Si TID = TID $\oplus N_R \oplus N_R$ tient, il authentifie le serveur et le lecteur.

Alors l'étiquette met à jour K i comme K_{i+1}= cro ($N_R \oplus N_S \oplus N_t$, K_i) et l'envoie au lecteur impliqué dans le message cro (RID $\oplus$ TID, K_{i+1}).

Sinon, l'authentification échoue.

Etape 7 :

Lors de la réception du message cro (RID $\oplus$ TID, K_{i+1}),

si cro (RID $\oplus$ TID, cro ($N_R \oplus N_S \oplus N_t$, K_i)) = cro (RID $\oplus$ TID, K_{i+1})

le lecteur met à jour K_i par la même équation K_{i+1} = cro ($N_R \oplus N_S \oplus N_t$, K_i) et l'envoie au serveur par le message cro (RID $\oplus$ TID, K_{i+1}).

Sinon, le protocole sera terminé.

Etape 8 :

Une fois que le serveur a reçu ce message, il effectue la même opération de vérification et s'il est en attente, le serveur met à jour K_i comme K_{i+1} = cro ($N_R \oplus N_S \oplus N_t$, K_i). Il calcule ensuite le message K_{i+1} TN T $\oplus N_R$ et l'envoie à le lecteur. Sinon, la connexion échoue

Etape9 :

A la réception du message $K_{i+1} \oplus N_t \oplus N_R$, si K_{i+1} = ($K_{i+1} \oplus N_t \oplus N_R$) $\oplus N_t \oplus N_R$ tient, le lecteur vérifie K_{i+1} et envoie le message $K_{i+1} \oplus N_t \oplus N_R$ à l'étiquette pour le même processus de vérification.

Autrement le protocole sera terminé.

etape10 :

Une fois que l'étiquette accepte la validité de K_{i+1} , elle marque Mark = 01, indiquant que la synchronisation de K_i est com- terminé.

Ensuite, l'étiquette calcule Mark $\oplus N_S$ et l'envoie au serveur via le lecteur. Notez que dans le travail original [2], Mark a été défini comme une chaîne de 2 bits alors que les paramètres comme N_S sont généralement plus grands chaînes, donc leur XOR n'a pas de sens. Cependant, sans perte de généralité, nous supposons que Mark est une extension triviale de cette chaîne de 2 bits.

etape11 :

Après avoir reçu le message Mark $\oplus N_S$, le serveur obtient la valeur de Mark et si elle est égale à 01, il conclut que la synchronisation de K i est terminée. Ensuite, le serveur ajoute un nouveau record cro (RID $\oplus$ TID, K_{+1i}), Rot ($K_{i+1} \oplus$ TID, $K_{i+1} \oplus$ RID) à IDT, après quoi la notification que l'enregistrement est complet la mise à jour est envoyée à l'étiquette via le lecteur.

etape12 :

Maintenant, l'étiquette définit Mark = 10, indiquant que le protocole d'authentification est terminé.

III.3.2 Le protocole HMNB :

Est un protocole d'authentification pour les RFID, il utilise une primitive cryptographique : la fonction de hachage [51] [52] [53]

Liste des notations utilisées :

La liste des notations utilisées dans le protocole fan et al est la suivante :

Notations	description
R, T	R : lecteur, T : étiquette.
TID	Identifiant prive du lecteur.
Nr, Nt	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
H	fonction de hachage.
ID	identificateur partage entre le lecteur et l'étiquette.
||	Opération de concaténation.
IDP	L'ancienne valeur de l'ID.
HID	Fonction de hachage de l'ID.

Tableau III.3: liste des notations utilisées dans le protocole HMNB.

Description du protocole

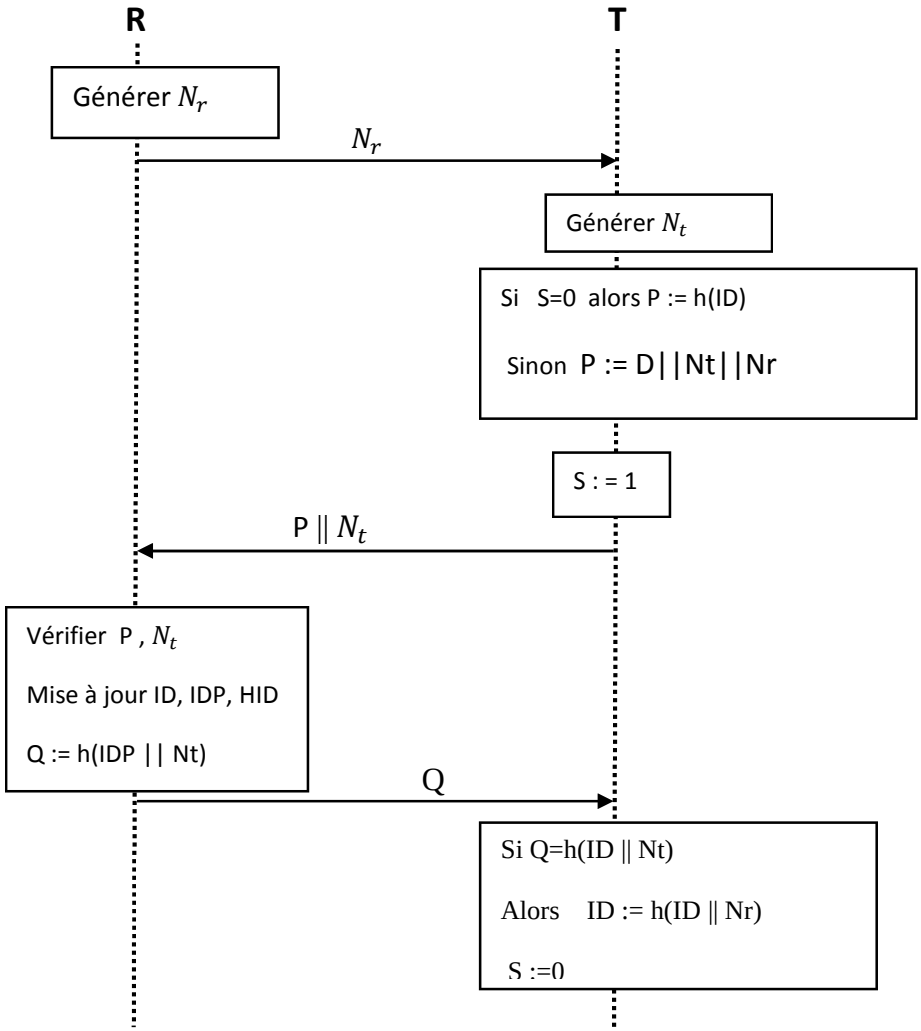


Figure III.3:schéma général du protocole HMNB

Etape 1 :

Le protocole est lancé par le lecteur : le lecteur génère un nombre aléatoire Nr et l'envoie a l'étiquette.

Etape 2 :

Apres avoir reçu Nr l'étiquette génère un nombre aléatoire Nt. Puis vérifie son état :

Si S=0 alors elle calcule P := h(ID)

Sinon calcule Sinon P := h(ID) || N_t || N_r

puis met son état a 1 (S = 1) et envoie P et Nt a l'étiquette.

Etape 3 :

A la réception de N_t et P le lecteur les vérifies puis

Il mets a jours ID, IDP, HID

Calcule Q := h(IDP || Nt)

Envoie Q a l'étiquette.

Etape 4 :

L'étiquette reçois Q et vérifie Q = H(ID||Nt), si oui T met à jour son ID puis remet son état à 0.

III.3.3 Le protocole RDW :

Liste des notations utilisées :

La liste des notations utilisées dans le protocole RDW est la suivante [51] [52] [53] :

Notations	description
R	R : lecteur,
T	T : étiquette.
Nr, Nt	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
K	Clé unique et partagée

Tableau III.4: liste des notations utilisées dans le protocole FDW.

Description du Protocol

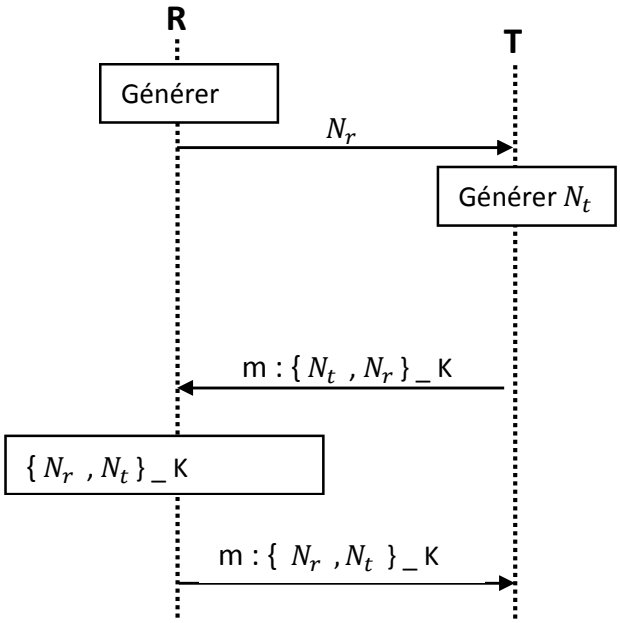


Figure III.4:Schéma général du protocole RDW

Dans ce protocole, chaque couple de lecteur R et tag T possède une clé unique et partagée K.

Etape 1 :

Le lecteur lance le protocole par l'envoi d'un nonce Nr au tag.

Etape 2 :

Le tag génère un nombre nonce Nt et crypte la paire (Nt, Nr) avec la clé partagée K, et l'envoie au lecteur.

Etape 3 :

Le lecteur déchiffre le message en utilisant la même clé partagée et inverse l'ordre des deux nonces, crypte le message avec la même clé partagée et l'envoie au tag.

La notation Alice-Bob pur ce protocole est:
R → T : Nr
T → R : {Nt,Nr}_K
R → T : {Nr,Nt}_K

III.3.4 Le protocole R^2AP (Reconstruction based RFID Authentiquassions Protocol) :

Est un protocole RFID ultraléger basé sur l'utilisation d'une nouvelle opération de reconstruction au niveau du bit. On note que la nomination des protocoles revient aux premiers caractères de nom des auteurs[50] .

Liste des notations utilisées :

Les notations utilisées dans le protocole R^2AP sont présentées dans le tableau suivant :

Notations	description
R	R : lecteur,
T	T : étiquette.
ID	Identificateur partagé entre le tag et le lecteur
IDS	Index de la table ou elle est stocké les secrets du tag
K1, K2, K3	Clés symétriques partagées entre le tag et le lecteur
N1, N2	Nombres aléatoires généré par le lecteur
Rot(x,y)	Rotation gauche de x par y bits
Rec(x,y)	L'opération de reconstruction de x par y
||	Opération de concaténation

Tableau III.5: liste des notations utilisées dans le protocole R^2AP.

Description du protocole

Ce protocole est basé sur l'échange des IDS, ID et les trois clés secrètes, désignées par K1,K2 et K3, et toutes les chaînes utilisées dans ce protocole doivent avoir une longueur l

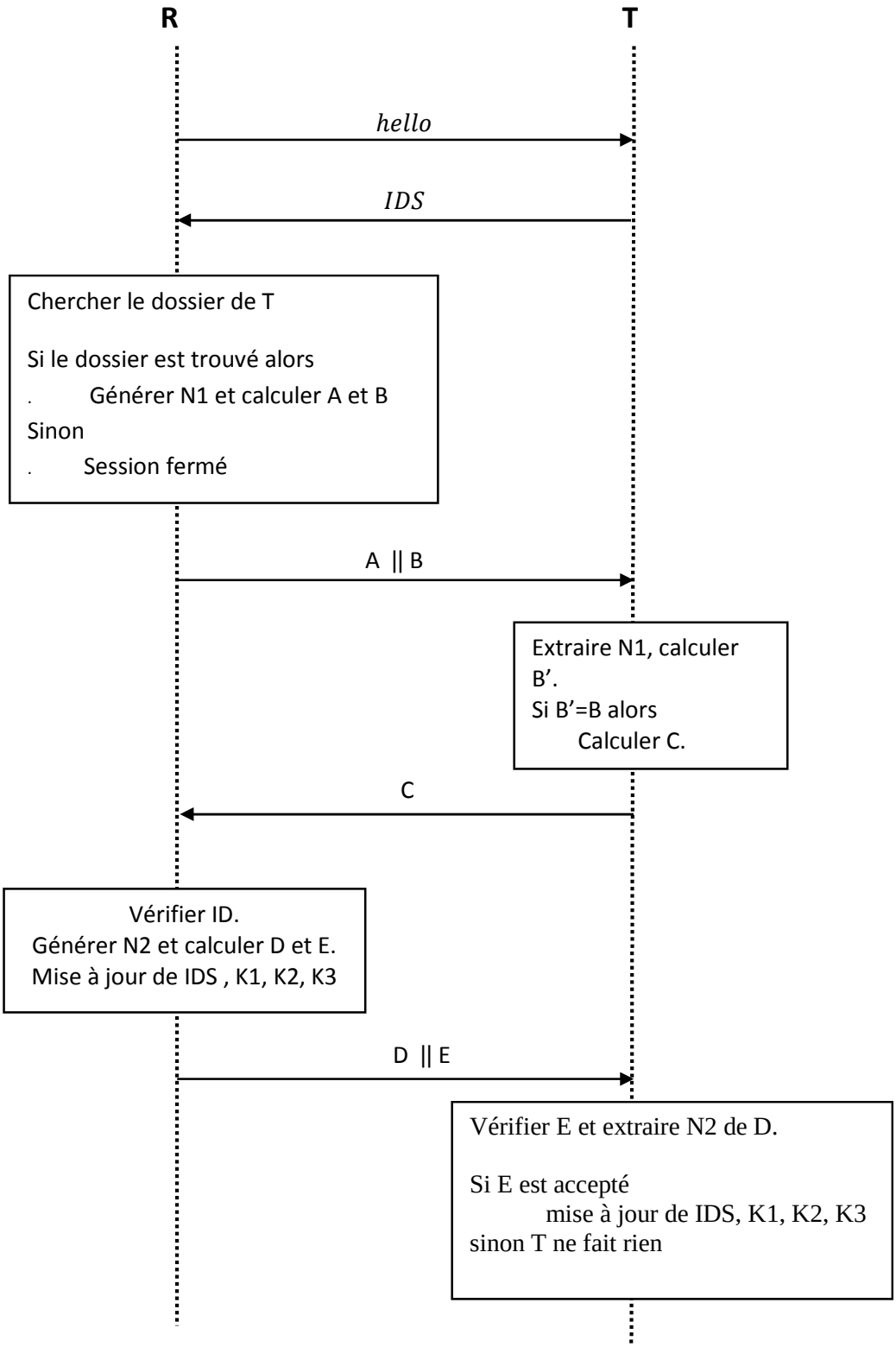


Figure III.5:schéma général du protocole R^2AP

Étape 1:
Le lecteur R envoie un message Hello au tag T pour initialiser une nouvelle session de protocole.

Étape 2:
T répond à R avec ses IDS.

Étape 3:
À la réception des IDS de T, R les utilisent comme un index de recherche des secrets du tag dans la base de données.
Si R trouve le dossier, l'étape 4 sera effectuée, autrement, R met fin à la session actuelle de protocole.

Étape 4:
R génère un nombre aléatoire N1, puis transmet des messages A et B à T, où

$A = Rec (K1 || K2) \oplus N1$ et
$B = Rot (Rec (K2 || N1), Rec (K3 || N1)) \oplus Rot (N1 || N1)$.

Étape 5:
Après avoir reçu les messages A et B, T extrait le nombre aléatoire N1 de message A et calcule ensuite un message B' en utilisant la formule de B avec les clés secrètes K1, K2, K3 et le nombre aléatoire N1 extrait.
Si B = B', T authentifie R comme un lecteur valide et transmet ensuite le message C comme une réponse, où: $C = Rec (Rec (K2 || K3), Rec (N1 || K1)) \oplus ID$.
Sinon, T met fin au protocole.

Étape 6:
Après avoir reçu C, le lecteur R peut authentifier T en utilisant l'ID extraite du message C.
Si l'ID correspond à celui de la base de données principale, R génère un nombre aléatoire N2 et calcule les messages D et E comme suit, et les transmet à T:

$D = R (N1 || K3) \oplus Rec (K1 || K3) \oplus N2$ et
$E = Rot (Rec (K2 || N2), Rec (K2 || N1)) \oplus Rot (N2 || N2)$.

Et puis, R mettra à jour ses secrets comme suit:
$IDSnew = Rec (I DS \oplus N2 || K3) \oplus K1$
$K1new = Rec (N2 || N1) \oplus K2$
$K2new = Rec (K2 || N1 \oplus N2) \oplus K3$
$K3new = Rec (K2 || K3) \oplus N1$

Étape 7:
Lors de la réception des messages D et E, T extrait le nombre aléatoire N2 du message D et teste la validité du message de E en utilisant ses clés secrètes. Si T accepte E, elle mit à jour ses secrets de la même manière que R. Sinon, T ne fait rien.

La notation Alice-Bob pour ce protocole est:

R → T: Hello
T → R: IDS
R → T: A || B
T → R: C
R T: D || E % tel que :
IDSnew = Rec (IDS ⊕ N2 || K3) ⊕ K1
K1new =Rec (N2 || N1) ⊕ K2
K2new = Rec (K2 || N1⊕ N2) ⊕ K3
K3new = Rec (K2 || K3) ⊕ N1

III.3.5 Le protocole EKE :

Liste des notations utilisées :

La liste des notations utilisées dans le protocole EKE est la suivante :

Notations	description
R, T	R : lecteur, T : étiquette.
Na , Ea, Nb	Nombre aléatoire généré par le lecteur et l'étiquette respectivement.
Kab	clé symétrique
K	clé de session

Tableau III.5: liste des notations utilisées dans le protocole EKE

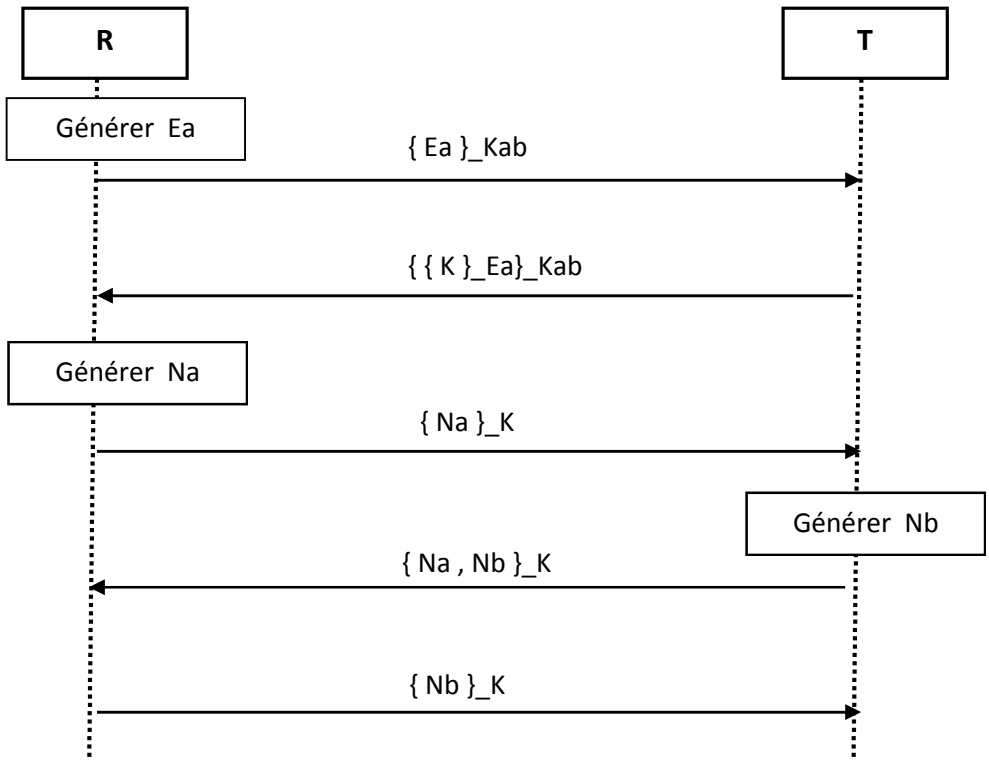


Figure III.6: le protocole EKE

La notation Alice-Bob pour ce protocole est:

A ⟶ B : { Ea }_Kab

B ⟶ A : { { K }_Ea}_Kab

A ⟶ B : { Na }_K

B ⟶ A : { Na , Nb }_K

A ⟶ B : { Nb }_K

Le protocole EKE, comme indiqué sur la figure (), exécute les étapes suivantes :

Etape 1 :
Le lecteur démarre le protocole en envoyant { Ea }_Kab (un nombre aléatoire E_a cripté par la cle symétrique Kab) à l'étiquette.

Etape 2 :
l'étiquette décrypte le message reçu pour obtenir Ea puis envois { { K }_Ea}_Kab (Ea chiffre par la clé de session K et la clé symétrique Kab.

Etape 3 :
Le lecteur envoie { Na }_K (un nombre aléatoire Na crypté avec la clé de session K) a l'étiquette

Etape 4 :

l'étiquette décrypte le message pour obtenir Na puis généré un autre nombre aléatoire Nb et envoi { Na , Nb }_K au lecteur

Etape 5 :
Le lecteur décrypte le message pour obtenir le Na et le compare au Na déjà généré : si ils correspondent donc l'étiquette est authentifié
Puis envoie { Nb }_K a l'étiquette

Etape 6 :
L'étiquette décrypte le message pour obtenir Nb puis le compare au Nb déjà généré si ils correspondent donc le lecteur est authentifié.

III.3.6. Schéma d'authentification proposé :

Nous proposons ici notre schéma de sécurisation d'un système RFID. Il utilise des opérations cryptographiques légères comme les PRNG, fonctions de hachage et XOR,
Dans notre schéma, nous utilisons des nombres aléatoires avec la valeur secrète de tag comme
Ensemencer et mettre à jour le secret de la balise après chaque authentification. La propriété de hasard des générateurs aide assurer la confidentialité et l'immunité contre les attaques par répétition 8 . Comme le secret est mis à jour après chaque session, transférez le secret est également assuré.

Hypothèse :

Nous supposons le canal de communication entre lecture et le serveur principal est entièrement sécurisé. Nous concentrons alors sur la sécurité de canal de communication entre le tag et le lecteur.
Le tag est un périphérique passif et communique avec le lecteur via un canal non sécurisé.
L'étiquette contient deux champs de données :

S : le secret de l'étiquette il est de 128 bits
ID : le pseudonyme d'étiquette (valeur d'index dans la base de données) est de 96 bits

Le serveur principal contient une base de données locale contenant des champs:

ID, h (ID) , $S_{nouveau}$

Liste des notations utilisées :

Notations	description
S,ID	Secret et ID pseudonyme de tag.
$S_{nouveau}$	Secrets de session actuels de la balise stockée dans le serveur principal
h ()	fonction de hachage.
PRNG(A)	pour trouver un nombre aléatoire avec A comme valeur de départ.
PRNG(A,B)	pour trouver un B.Nombre pseudo_aléatoire avec A comme valeur de départ.
||	Fonction de concaténation de chaines.
⊕	La fonction XOR.

Tableau III.6: liste des notations utilisées dans le protocole proposé

Description du Protocole :

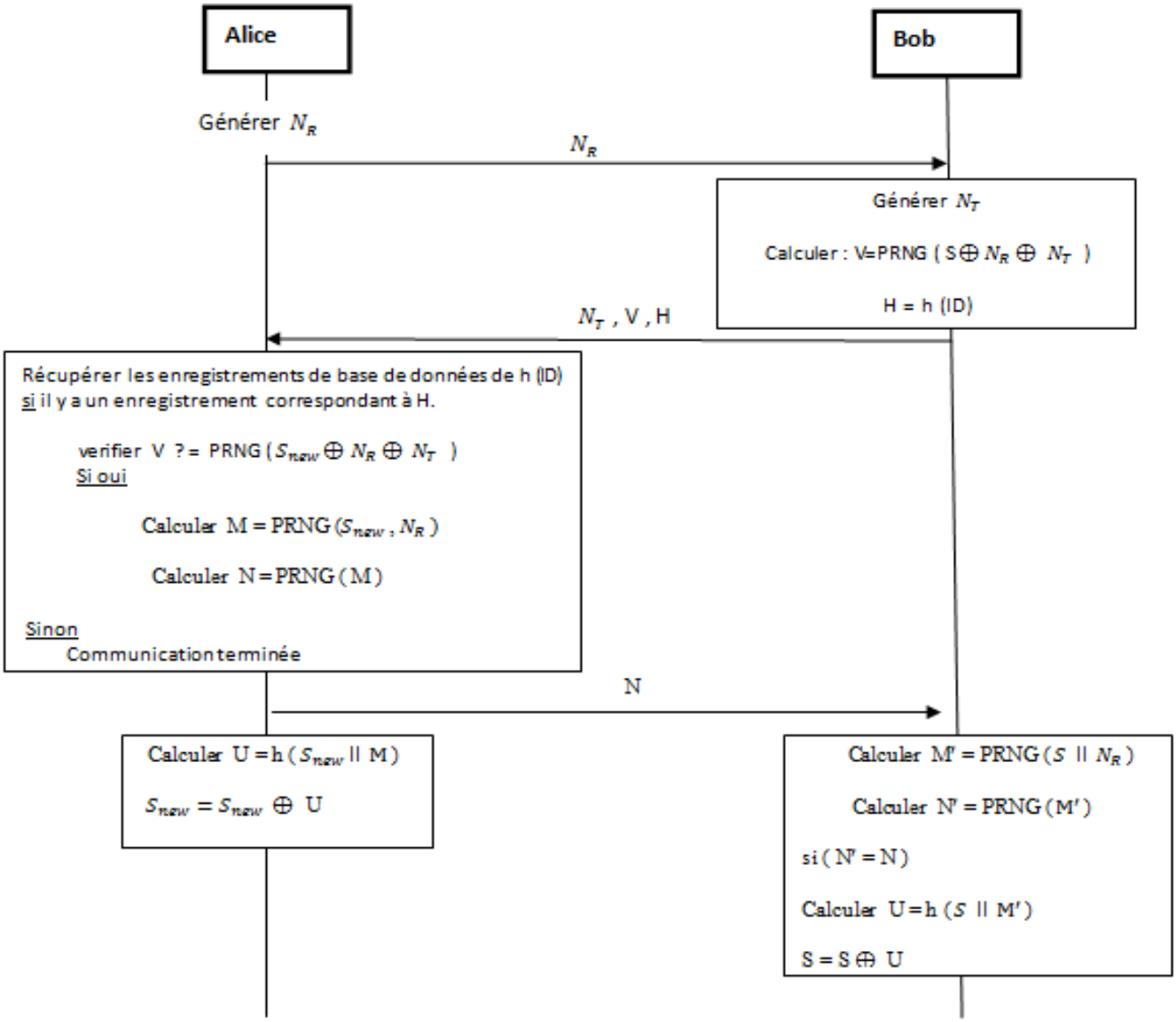


Figure III.7: Schéma générale de protocole proposé

Ce protocole exécute les étapes suivantes :

Étape 1:
Le lecteur initie le processus d'authentification en interrogeant le tag puis sélectionne un tag et demande ses informations.
Le lecteur génère un nombre aléatoire (N_r) et l'envoie à l'étiquette .

Étape 2:
Après avoir reçu $N_r$du lecteur, l'étiquette génère un autre nombre aléatoire (N_t). puis calcule V = PRNG (S $\oplus N_R \oplus N_t$).) et H = h (ID), en utilisant les informations (Tag ID, S) stockées dans l'étiquette.
L'étiquette envoie V, H et N_t au lecteur.

Étape 3:
Le lecteur après avoir reçu ces valeurs, récupère les enregistrements de base de données de h (ID) pour trouver s'il y a un enregistrement correspondant à H.

• Si aucun enregistrement ne correspond à cette valeur, la communication est terminée.
• Si un enregistrement est trouvé, le serveur extrait le secret S New de la balise

correspondante de la base de données et calcule V ' = PRNG (S Nouveau $\oplus N_r \oplus N_t$) pour vérifier si V ' et le V reçu sont identiques ou non.

Si elles sont égales,c'est confirmé que la session précédente a été couronnée de succès et tag contient S new comme valeur S.

Étape 4:
En utilisant M comme valeur de départ, il calcule un nombre aléatoire (N) à partir de PRNG.
Le lecteur envoie N au l'etiquette.
La même chose doit être faite au niveau de l'étiquette

Étape 5:
Pour confirmer que le lecteur est authentifié, l'étiquette calcule M ' =PRNG(S || N_r) le nombre pseudo-aléatoire en prenant S (secret stocké dans l'étiquette) comme valeur initial.

En utilisant M 'comme valeur de départ, il calcule un nombre aléatoire (N ') à partir de PRNG.
La balise vérifie si N ' et N sont identiques. Si oui, l'étiquette est confirmée que l'information provient du lecteur légitime. L'étiquette calcule U = h (S || M ') et met à jour sa valeur secrète en tant que S⊕U.

Étape 6:
Le lecteur calcule U = h (S New || M).
Puis met à jour la valeur secrète dans la base de S new = S new ⊕ U.

III.4. Conclusion :

Ce chapitre est consacré aux protocoles cryptographiques dans les systèmes RFIDs, dans on a présente quelques protocoles d'authentification et dans le chapiste suivant on verra la vérification automatique de la sécurité des protocoles cryptographiques avant leurs mises en service en utilisant des outils automatisés tout en concentrons sur l'outil AVISPA qui permet la spécification, l'analyse et la validation des protocoles de sécurité

Chapitre IV :

La vérification formelle d'un protocole RFID a laide D'AVISPA et SPAN

IV.1 Introduction :

La vérification de la sécurité des protocoles cryptographiques avant leurs mises en service est très importante car leur sécurité n'est pas garantie par l'usage des méthodes de chiffrement seulement, mais aussi par une vérification automatique.

Nous présentons donc dans ce chapitre quelques notions relatives à la vérification, en se basant sur le modèle formelle, ces caractéristiques, ces méthodes et les outils de vérification automatique qui s'appuient sur ce modèle. Nous concentrons sur l'outil AVISPA qui permet la spécification, l'analyse et la validation des protocoles de sécurité.

Ensuite on verra le langage formel HLPSL (High Level Protocol Specification Language) qui permet la spécification modulaire afin de vérifier les propriétés de sécurité propose dans le chapitre 3 à l'aide de l'outil AVISPA .

IV.2 Les notions de base de la vérification :

La vérification est l'étape qui permet de tester, de prouver et de confirmer qu'un protocole est sûr ou non :

Les protocoles : Les protocoles cryptographiques sont des petits programmes qui spécifient une séquence d'émissions/réceptions de messages qui visent à établir entre deux ou plusieurs participants (agents), des communications répondant à certaines propriétés de sécurité

Les agents : Il y a deux types d'agents ou participants :

Participants honnêtes *:*
Ce sont Les agents qui ont un comportement qui suit celui énoncé par une exécution normale du protocole

Intrus :
Un intrus ou attaquant est un participant qui ne suit pas exactement le déroulement du protocole. Il espionne les communications qui circulent entres les agents, joue plusieurs sessions de protocoles avec des participants, en se faisant passer pour un agent honnête et ainsi effectue des actions non prévues par la spécification du protocole, afin de découvrir des informations supposées rester secrètes.

L'attaque :
Une attaque est l'exécution d'une ou plusieurs sessions du protocole qui permet à l'intrus d'apprendre une information supposée secrète. On peut modéliser une attaque en utilisant un outil automatique de vérification de protocoles cryptographiques comme AVISPA .

Secret :
Un secret est une donnée confidentielle ne devant pas être découverte par une tierce personne qui n'est pas censée la connaitre. Un protocole vérifie la propriété de secret, si une personne malhonnête (l'intrus) en fonction de ses capacités ne peut jamais obtenir les données échangées entre plusieurs participants honnêtes.

IV .3 Les objectifs de la vérification :

La vérification de la sécurité des protocoles cryptographiques dépend généralement de deux axes complémentaires : la recherche d'une attaque et la preuve d'un protocole sûr [57] .

IV.4 La vérification formelle :

La vérification formelle est une méthode qui permet la description mathématique d'un protocole et les propriétés de sécurité, ainsi les étapes à suivre pour déterminer si le protocole satisfait ces propriétés (si le protocole est sécurisé) [58].

Avant de faire l'étude des protocoles de sécurités, il est nécessaire de les modéliser, à l'aide de cette modélisation on feras une vérification formelle qui nous permet de conclure si le protocole ne révèle en aucun cas une certaine faille ou qu' il assure certaines propriétés.

A la fin de la vérification la détection des attaques est possible ou, au moins, des points faibles que la spécification d'un protocole peut comporter.

L'objectif de modèle symbolique est le fait qu'il permet d'obtenir des preuves mathématiques simplifiées et souvent automatisées.

IV.4.1 Les concepts de base relatifs aux modèles symboliques utilisés dans ce mémoire :

Les messages :

Dans les modèles symboliques, les messages sont généralement représentés par des termes, pour cette raison, la définition de modèles symboliques passe par la définition d'une algèbre de termes. Par exemple, si k et m sont des termes, le chiffrement symétrique de m par k est noté par {m}_k ,Les primitives cryptographiques sont donc représentées par des symboles.

Les primitives cryptographiques :

Les primitives cryptographiques comme la fonction de hachage et les clés asymétriques, sont considérées comme des boîtes noires, c'est-à-dire qu'elles sont sûres par définition et qu'il n'existe pas d'algorithme pour les casser.

Les systèmes de transition :

L'exécution d'un protocole est modélisée par une séquence finie d'états globaux et des transitions entre ces états globaux, un état global contient tout les informations sur les messages échangés et les états locaux des agents. L'état local d'un agent est comparable à l'état de son mémoire à un certain moment d'exécution du protocole. Nous obtenons ainsi une trace d'exécution d'un protocole qui est une suite alternante d'états globaux et de transitions entre ceux-ci.

Le canal de communication :

Le but de canal de communication sert à l'échange des messages entre les agents. On distingue deux types de canaux de communications : les canaux de communications publics et les canaux de communications privés :

Les canaux de communication publics : un canal de communication est dit public si Les messages échangés sur ce canal peuvent être vus par tous les participants qu'il soit honnête ou non c'est a dire que ces messages sont connus de tous, en particulier de l'intrus qui est capable d'initialiser le protocole avec d'autres participants, il peut intercepter, rejouer ou modifier les messages pendant l'exécution.

Les canaux de communication privés : dans ce type de de canaux de communication seuls ces participants honnêtes peuvent recevoir et envoyer des messages. Par conséquent, un intrus ne peut pas donc écouter les messages qui circulent sur ce genre de canaux.

IV.4.2 Les méthodes de vérification formelle :

Il existe trois principales méthodes de la vérification formelle de protocoles : La méthode du raisonnement logique, les méthodes d'exploration de l'état et les méthodes de démonstration de théorèmes[59] .

La méthode du raisonnement logique :
C'est la première méthode utilisé dans la vérification automatique des protocoles parmi ses techniques essentielles on cite :

- ***La BAN*** : C'est la plus importante technique de raisonnement, elle décrit les principes de communication, messages envoyés et reçus tout au long du processus d'exécution du protocole. Ces raisonnements logiques aboutissent à une assertion finale qui permet de juger si un protocole est correct ou pas.

- ***La GNY (extension de la BAN)***.Plus sophistiquée et complexe, elle améliore la BAN en mettant en évidence la différence entre le contenu d'un message et sa signification .

- ***La BGNY (extension du GNY)*** : Elle permet de spécifier les propriétés du protocole à des niveaux intermédiaires (insertion de l'opération de hachage, algorithmes d'échange de clés…)

L'approche par preuve de théorèmes
Cette approche est basée sur un raisonnement mathématique pour démontrer l'exactitude d'un protocole de sécurité, Les outils développés pour cette méthode comme ACL2, Isabelle, utilisent respectivement la FOL (First Order Method) et la HOL (High Order Method). Cette approche permet d'obtenir une preuve formelle, mais n'est pas totalement automatisée
Les méthodes d'exploration d'états
Ces méthodes basées sur la construction d'un modèle de protocole et de vérifier que chaque état atteignable satisfait certaines propriétés.
Ce type de méthodes peuvent être divisées en deux catégories :

La vérification non-bornée : Dans cette approche, on recherche toutes les exécutions

possibles du protocole, en vérifiant chaque état accessible qui satisfait certaines conditions, ,elle considère une infinité de principes et une infinité de sessions. L'espace des états généré par l'analyse d'un protocole peut être infini, on parle alors d'explosion d'états.

La vérification bornée : Dans cette méthode le nombre de sessions est fixé, La méthode de vérification recherche alors un état particulier où les propriétés de sécurité sont violées et génère un contre-exemple en exploitant une trace depuis cet état jusqu'à un état initial. C'est sur cette approche que reposent tous les outils de vérification automatique de protocoles.

IV.4.3 outils de vérification formelle :

Dans le domaine de vérification automatique des protocoles de sécurité, Il y a plusieurs analyseurs de protocoles ,mais la plateforme AVISPA (Automated Validation of Internet Security Protocols and Applications) est l'analyseur le plus connu qui modélise un grand nombre de protocoles. L'efficacité d'Avispa a été testée sur de nombreux protocoles récemment standardisés, par exemple par l'IETF (Internet Engineering Task Force) et des protocoles du domaine e-business.
Cet outil est compatible avec loutil graphyqique SPAN et possède une syntaxe simple par rapport aux autre outils . pour ces raison nous l'avons utilisé dans notre projet[59] .

La plateforme AVISPA :

En juillet 2005 les partenaires du projet européen AVISPA ont publié leurs travaux de développement d'une plateforme contenant quatre outils d'analyse de protocoles et permettant la détection des attaques logiques sur les protocoles de sécurité. Cette plateforme suggère aussi des améliorations assurant la validité des propriétés de confidentialité et d'authentification [59] La structure de l'outil AVISPA est représenté sur la Figure suivante :

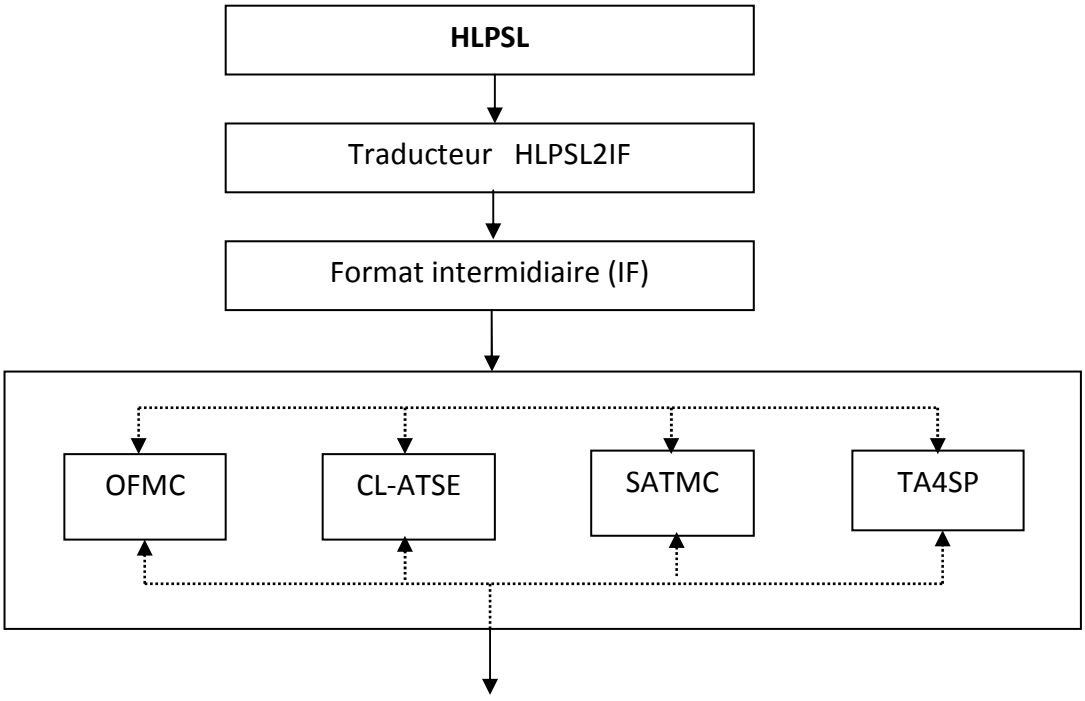


Figure IV.1 : La structure de l'outil AVISPA

L'outil supporte quatre moteurs de vérifications dis "backends" :

The On-the-fly Model-Checker (OFMC): Utilise plusieurs techniques symboliques pour explorer l'espace d'états à la demande.

CL-AtSe (Constraint-Logic-based Attack Searcher): Applique la résolution, de contraintes avec heuristiques de simplification et techniques d'élimination de redondances

The SAT-based Model-Checker (SATMC): Construit une formule propositionnelle qui encode toutes les attaques possibles (de longueur limitée) sur le protocole et soumet le résultat à un solveur

SAT TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols): Estime les connaissances de l'intrus en utilisant un langage d'arbre régulier avec réécriture pour produire des sous et sur-approximations

Une spécification HLPSL est traduite au format intermédiaire (IF), en utilisant un traducteur appelé hlpsl2if. Notez que cette étape de traduction intermédiaire est transparente pour l'utilisateur

Environnement graphique

AVISPA offre également un environnement graphique de mise au point des protocoles comme illustré sur la figure suivante :

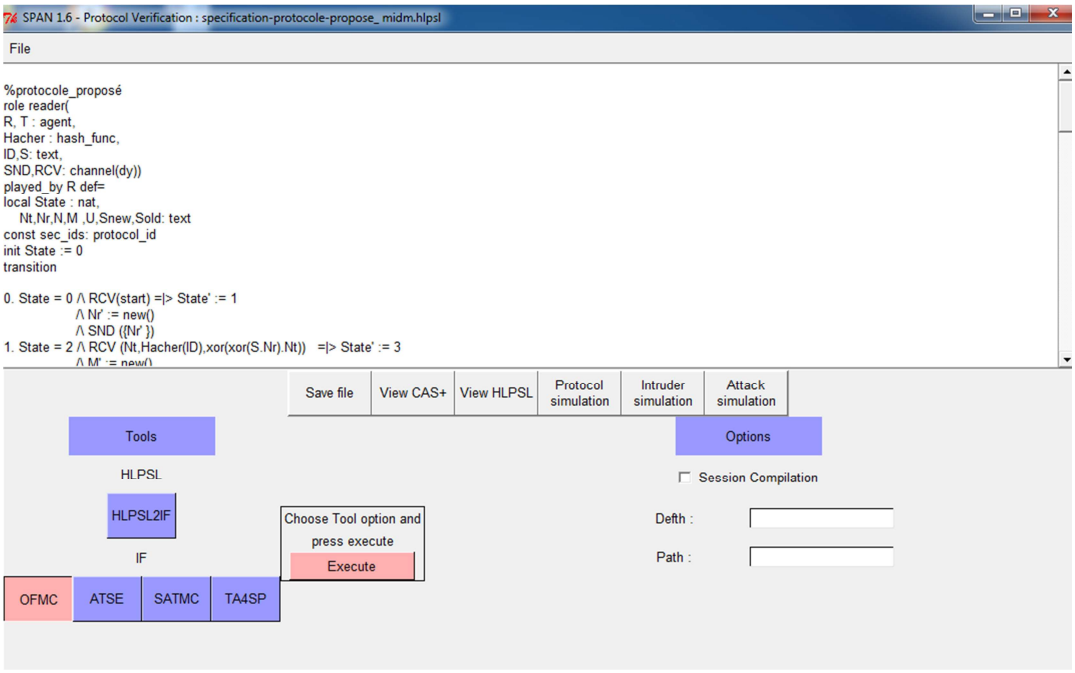


Figure IV.2 : Environnement Graphique d'AVISPA

Le langage HLPSL :

HLPSL (High Level Protocol Specification Language) est un langage formel de spécification modulaire du protocole de haut niveau basé sur des descriptions de rôles. Il supporte des primitives cryptographiques différentes (Clés symétriques et asymétriques, les fonctions de hachage) et de leurs propriétés algébriques (ou exclusif, exposant).

Le but de ces spécifications étant de pouvoir vérifier des propriétés de sécurités, l'authentification et la confidentialité. L'idée principale est de représenter un protocole cryptographique par un système d'états/transitions pour lequel il est possible de vérifier des propriétés de sécurité exprimées en logique temporelle linéaire (LTL).
Les transitions définissent le comportement du protocole de sécurité, et ainsi, à partir de l'état initial, nous sommes capables d'énumérer les états atteignables du protocole étudié.

Les spécifications HLPSL de protocoles sont divisées en rôles, ces derniers sont répartis en deux catégories distinctes :

Les rôles basiques : représente les agents participants aux protocoles
Les rôles composition : représente les scénarios des rôles basiques.

A la fin de la spécification, on détermine les propriétés de sécurité à vérifier.

Structure d'une spécification HLPSL

Une spécification HLPSL d'une communication entre deux agents Alice & Bob est structurée comme suite (à noter en HLPSL le symbole % signifie début de commentaire): On commence toujours par la spécification des rôles basiques, ici rôle Alice et rôle Bob[59]

role Alice (arguments)

played_by alice

----- }
----- } Liste des déclarations
----- }
----- }

Transition

----- }
----- } Les transitions de ''Alice''
----- }

end role

role Bob (arguments)

played_by Bob

----- }

----- Liste des déclarations

Transition

----- } Les transitions de ''Bob''
-----

end role

% on passe maintenant à la spécification des rôles de composition: rôle session et rôle %envirenment

role session (arguments)

----- } Listes des déclarations
-----

composition

Alice(les arguments) /\ Bob (les arguments) } composition de la session de communication entre Alice et Bob

end role

role environment( )

----- } Liste des déclarations et les connaissances initiales de l'intrus
-----

Composition

session(arguments) /\ session(argument) ----- } Établissement des sessions de communication entre Alice, Bob et intrus (spécification d'attaque)

end role

% On passe maintenant à la déclaration des propriétés à vérifier: goal

Goal

} Les propriétés à vérifier

end goal

environment()

Caractéristiques de HLPSL

Le langage HLPSL possède plusieurs caractéristiques, citons:

- support cryptographiques variés (clés symétriques, clés publiques, fonctions de hachage).
- information typée (ou non), avec des types simples ou composés.
- propriétés algébriques supportées (concaténation, OU exclusif, exponentiation).
- canaux pour les échanges de messages.

Comme on remarque, le langage de spécification HLPSL ne supporte pas quelques caractéristiques importantes comme:

- Les opérateurs arithmétiques: +, -, *, / …
- L'opérateur logique: ou (\/)
- Les fonctions de décalages, rotations, permutation ….
- Les boucles : if…else, while, repeat
- Le choix: Case.

IV.5.La vérification formelle du protocole proposé :

Dans cette section, on vérifie les propriétés de la confidentialité de l'identificateur ID et Le S (sec_ID et sec_S respectivement), l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader) du protocole propose décrit dans le chapitre III.

Ces propriétés de securite a verifier sont spécifiées dans HLPSL au niveau de la partie goal comme suit:

```
goal
      secrecy_of sec_id, sec_s
      authentication_on aut_tag
      authentication_on aut_reader
end goal
```

la spécification des rôles basiques :

```
%protocole_proposé
role reader(
R, T : agent,
Hacher : hash_func,
PRNG:function,
ID,S: text,
SND,RCV: channel(dy))
played_by R def=
local State : nat,
  V,H, Nt,Nr,N,M ,U,Snew: text
const sec_s: protocol_id
init State := 0
transition

0. State = 0 /\ RCV(start) =|> State' := 1
              /\ Nr' := new()
              /\ SND (Nr' )
             /\ secret(Nr,sec_s,{R,T})


1. State = 2 /\ RCV (Nt,H,V)   =|> State' := 3
              /\ V':= PRNG(xor(xor(S.Nr).Nt))


2. State = 2 /\ equal (V,V')   =|> State' := 5
              /\ M' := PRNG(Snew.Nr)
              /\ N' := PRNG(M)
              /\ SND (N)


3. State = 4 /\ RCV (start) =|>  State' := 7
              /\ U' := Hacher( Snew.M )
              /\ Snew' := xor(Snew,U)
```

```
end role


role tag ( T,R: agent,
Hacher : hash_func,
PRNG:function,
ID,S:text,
SND,RCV: channel(dy))
played_by T def=
local State : nat ,
    Nt,Nr,N,M ,U,Snew,V,H: text
const sec_id: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(Nr) =|> State' := 1
                /\ Nt' := new ()
                /\ V' := PRNG(xor(xor(S.Nr).Nt))
                /\ H' := Hacher(ID)


2. State = 2 /\ RCV (start) =|> State' := 3
                /\ SND(H,V,Nt)
                /\ RCV(N)
                /\ secret(Nt,sec_id,{T,R})


3. State = 4 /\ RCV (start) =|> State' := 5
                /\ M' := PRNG(S.Nr)
                /\ N' := PRNG(M)


4. State = 6 /\ equal(N,N') =|> State' := 7
                /\ U' := Hacher( S.M )
                /\ S' := xor(S , U)
end role
```

la spécification des rôles de composition: rôle session et rôle environment

```
role session(T,R : agent,Hacher: hash_func ,PRNG:function,ID,S:text) def=
local St,Rt,Sr,Rr : channel(dy)

composition
tag(T,R,Hacher,PRNG,ID,S,St,Rt) /\ reader(R,T,Hacher,PRNG,ID,S,Sr,Rr)
end role
```

Concernant l'authentification, il y a deux attaques possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle. Cela est spécifié dans le rôle environnement dans HLPSL comme suit :

Ataque Main-in-the-middle :

Le scenario spécifié dans le rôle environnement ci-dessous permet de détecter des attaques du type Main-in-the-midlle s'il existe :

```
role environment() def=
const t,r : agent,
hach: hash_func,
prng:function,
id,idi,si,s: text,

aut_tag, aut_reader:protocol_id
intruder_knowledge = {t,r,hach,prng,idi,si}
composition

%attaque MITM
session(t,r,hach,prng,id,s)   /\  session(t,i,hach,prng,idi,si) /\  session(i,r,hach,prng,idi,si)
end role

environment()
```

Après la vérification de ce protocole par l'outil CL-AtSe d' AVISPA, le résultat est comme Suit **:**

```
% Version of 2006/02/13

SUMMARY

  SAFE

DETAILS

  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL

  C:\progra~1\SPAN\testsuite\results\protocole propose ( attaque MITM).if

GOAL

  as_specified

BACKEND

  OFMC

COMMENTS

STATISTICS

  parseTime: 0.00s
```

searchTime: 0.03s

visitedNodes: 4 nodes

depth: 2 plies

Ce résultat signifie qu'il n y a pas d'attaque Main-in-the-Middle.

Attaque par rejeu :

Dans l'attaque par rejeu (Replay Attack), l'adversaire peut écouter le message de réponse du tag et du lecteur. Il retransmettra le message écouté sans modification au lecteur plus tard. La spécification ci-dessous du rôle environnement en HLPSL dépend du traitement de deux sessions en parallèles (représenter par le symbole /\ ) entre deux tags légitimes et un même lecteur (t1, t2 et r). Ce scenario permet de détecter les attaques du type "Attaque par rejeu" s'elles existent.

```
role environment() def=
const t1,t2,r : agent,
hach: hash_func,
prng:function,
idt1,idst1,idt2,idst2: text,

aut_tag, aut_reader:protocol_id
intruder_knowledge = {t1,t2,r,hach,prng}
composition

%attaque par rejeu
session(t1,r,hach,prng,idt1,idst1)   /\  session(t2,r,hach,prng,idt2,idst2)

end role

environment()
```

Après la vérification de ce protocole ,le résultat est comme suit:

```
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\specification-protocole-propose_ midm.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 4 nodes
```

depth: 2 plies

Ce résultat signifie qu'il n y a pas d'attaque par rejeu.

IV.6 La vérification formelle du protocole EKE :

Dans cette section, on vérifie les propriétés de la confidentialité, l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader) du protocole EKE décrit dans le chapitre III.
Ces propriétés sont spécifiées dans HLPSL au niveau de la partie goal comme suit:

```
end role

goal

 secrecy_of sec_k1, sec_k2
 authentication_on nb
 authentication_on na

end goal
```

la spécification des rôles basiques :

```
role alice (A,B: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by A
def=

  local State   : nat,
      Ea,Na,Nb,K: text

 const sec_k1 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\ Rcv(start) =|>   State' := 1
              /\ Ea' := new()
              /\ Snd({Ea'}_Kab)

   2. State = 1  /\ Rcv({{K'}_Ea}_Kab) =|>  State' := 2
               /\ Na' := new()
               /\ Snd({Na'}_K')
              /\ secret(K',sec_k1,{A,B})
               /\ witness(A,B,na,Na')

   3. State = 2 /\ Rcv({Na.Nb'}_K)  =|>  State' := 3
              /\ Snd({Nb'}_K)
              /\ request(A,B,nb,Nb')
```

```
end role


role bob (B,A: agent,
          Kab: symmetric_key,
          Snd,Rcv: channel(dy))
played_by B
def=

  local State   : nat,
      Ea,Na,Nb ,K: text

const sec_k2 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\ Rcv({Ea}_Kab) =|> State' := 1
            /\ K' := new()
            /\ Snd({{K'}_Ea}_Kab)
           /\ secret(K',sec_k2,{A,B})

   2. State = 1 /\ Rcv({Na}_K) =|> State' := 2
            /\ Nb' := new()
            /\ Snd({Na'.Nb}_K)
            /\ witness(B,A,nb,Nb)

   3. State = 2 /\ Rcv({Nb}_K)=|>  State' := 3
            /\ request(B,A,na,Na)

end role
```

la spécification des rôles de composition: role session et role envirenment :

```
role session(A,B: agent,
          Kab: symmetric_key)
def=

  local SA, RA, SB, RB: channel (dy)

  composition
    alice(A,B,Kab,SA,RA) /\ bob(B,A,Kab,SB,RB)

end role
```

Concernant l'authentification, il y a deux attaques possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle. Cela est spécifié dans le rôle environnement dans HLPSL comme suit :

Attaque par rejeu :

. La spécification ci-dessous du rôle environnement en HLPSL permet de détecter les attaques du type "Attaque par rejeu" si elles existent.

```
role environment()
def=

  const a, b ,r   : agent,
        kab    : symmetric_key,
        na, nb : protocol_id,
kt11,kt21:symmetric_key

  intruder_knowledge={a,r}

  composition

%attaque par rejeu

session(a,r,kt11)   /\  session(b,r,kt21)

end role
```

Après la vérification de ce protocole par l'outil OFMC d' AVISPA, le résultat est comme :

```
suit:
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\EKE par rejeu.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.15s
  visitedNodes: 49 nodes
  depth: 12 plies
```

Ce résultat signifie qu'il n y a pas d'attaque par rejeu.

Attaque Main-in-the-middle :

Le scenario spécifié dans le rôle environnement ci-dessous permet de détecter des attaques du type Main-in-the-midlle s'il existe :

```
role environment()
def=

  const a ,r   : agent,
       kab    : symmetric_key,
       na, nb : protocol_id,
k:symmetric_key
intruder_knowledge={a,r}

  composition

%attaque MITM

session(a,r,k)   /\  session(a,i,k) /\  session(i,r,k)

end role
```

Après la vérification de ce protocole par l'outil OFMC d' AVISPA, le résultat est comme suit:

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\EKE MIDM avec attaque.if
GOAL
  authentication_on_nb
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.01s
  searchTime: 0.07s
  visitedNodes: 29 nodes
  depth: 4 plies
```

Ce résultat signifie que le Protocol EKE est vulnérable à l'attaque man in the middle ,et l'attaque du lecteur (authentication_on_nb)

IV.7 Proposition d'une amélioration du protocole EKE :

Comme nous l'avons vérifié formellement, le protocole EKE ne résiste pas a l'attaque d'authentification main in-the-middle .

Dans cette section, on va présenter une amélioration du protocole EKE. Pour laquelle on vérifie la confidentialité, l'authentification du tag et l'authentification du lecteur par les outils AVISPA et SPAN. Le protocole conçu résiste a l'attaque main in-the-middle.
En comparant l'ordre des messages transmis dans le protocole EKE avec celui du protocole proposé et vérifie auparavant on remarque que authentification du tag auprès du lecteur est faite au départ comme suit :

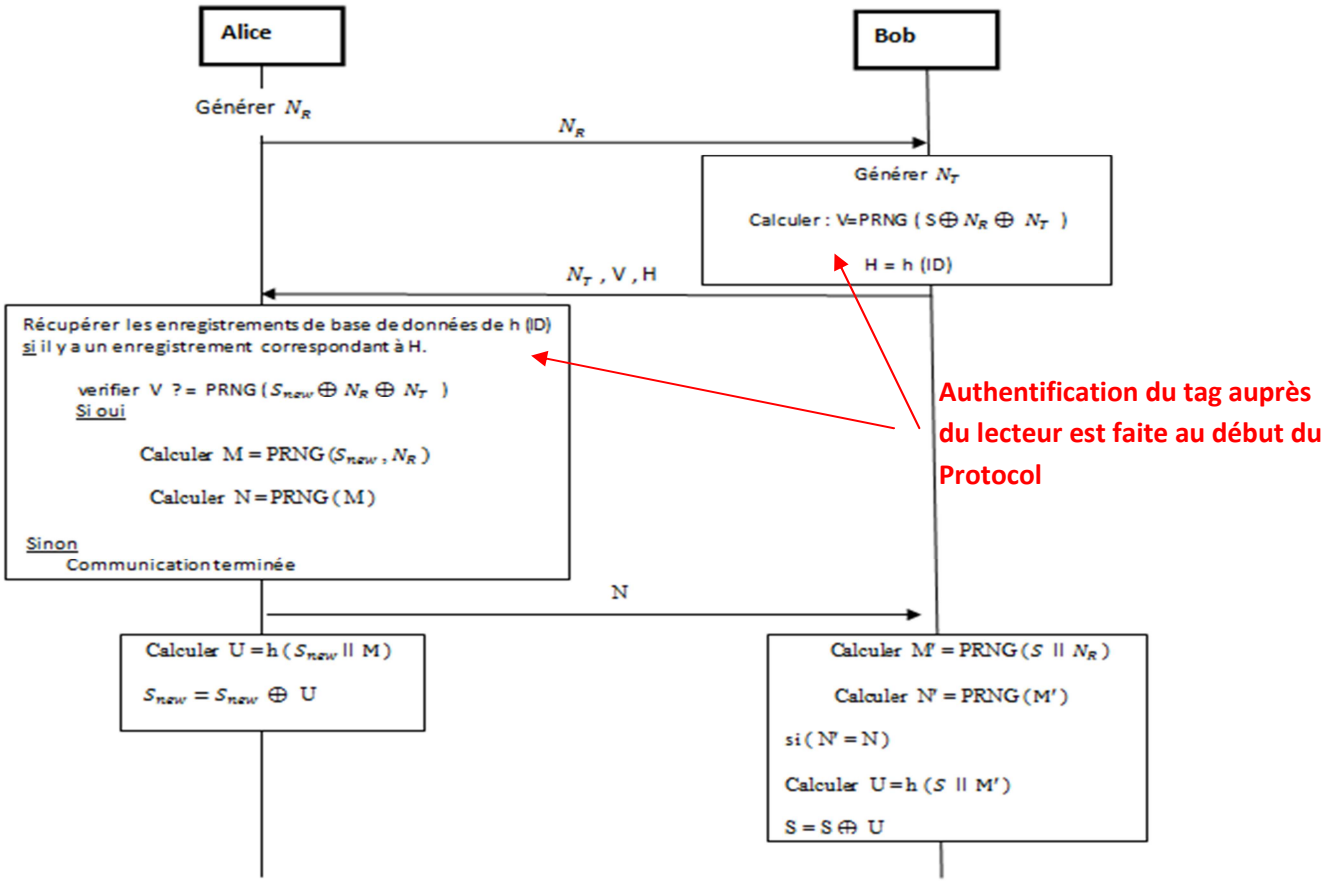


Figure IV.3 : Schéma générale de protocole proposé

Tandis que dans le protocole EKE l'authentification du tag est faite a la fin du protocole dans l'étape 5

Etape 5 :

Le lecteur décrypte le message pour obtenir le Na et le compare au Na déjà généré : si ils correspondent donc l'étiquette est authentifié
Puis envoie { Nb }_K a l'étiquette :

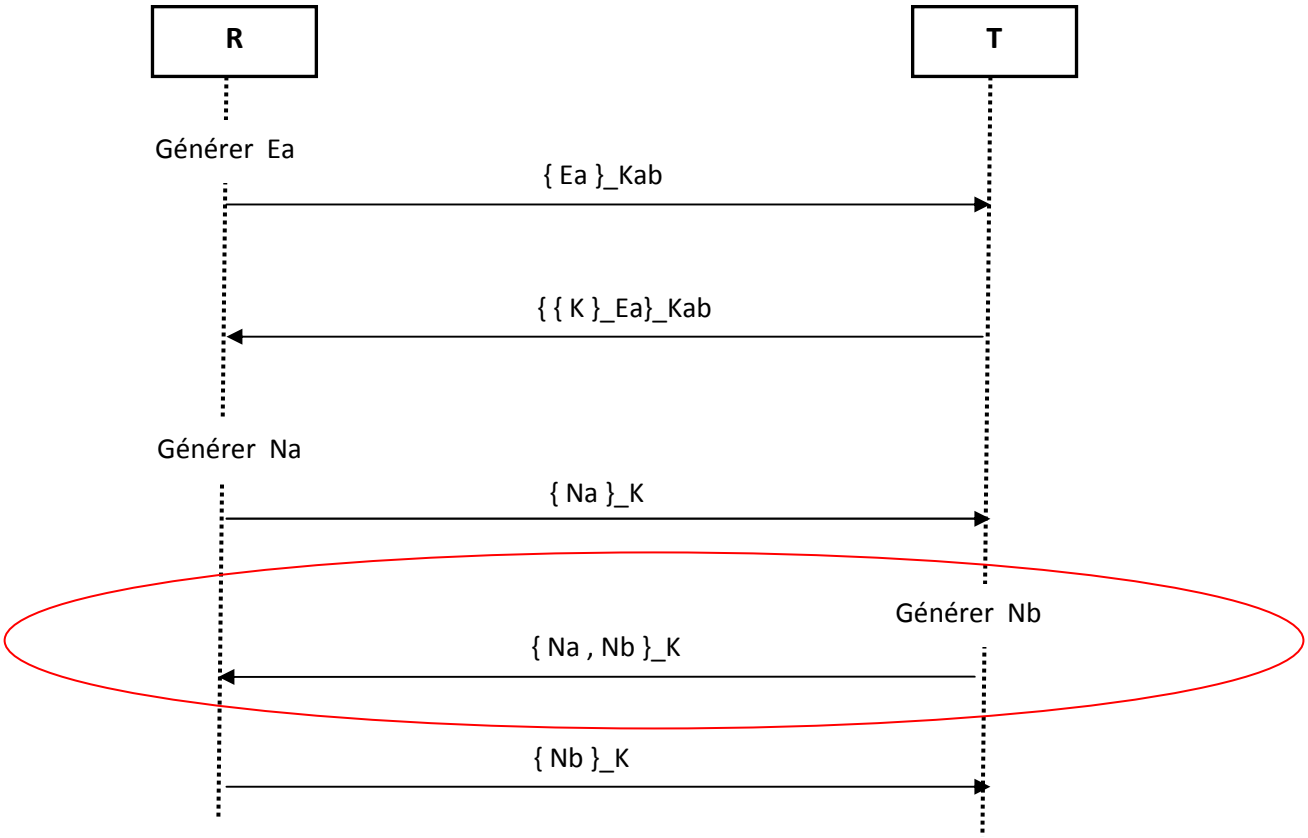


Figure IV.4: le protocole EKE

Pour le protocole EKE on propose l'authentification du tag auprès du lecteur juste au début de la communication : le lecteur reçoit l'identificateur du tag et le compare a celui déjà enregistre dans la base de données :

- s'il y a un enregistrement correspondant à Id-tag alors le tag est authentifié.
- Sinon communication sera terminée.

Comme le montre le schéma suivant :

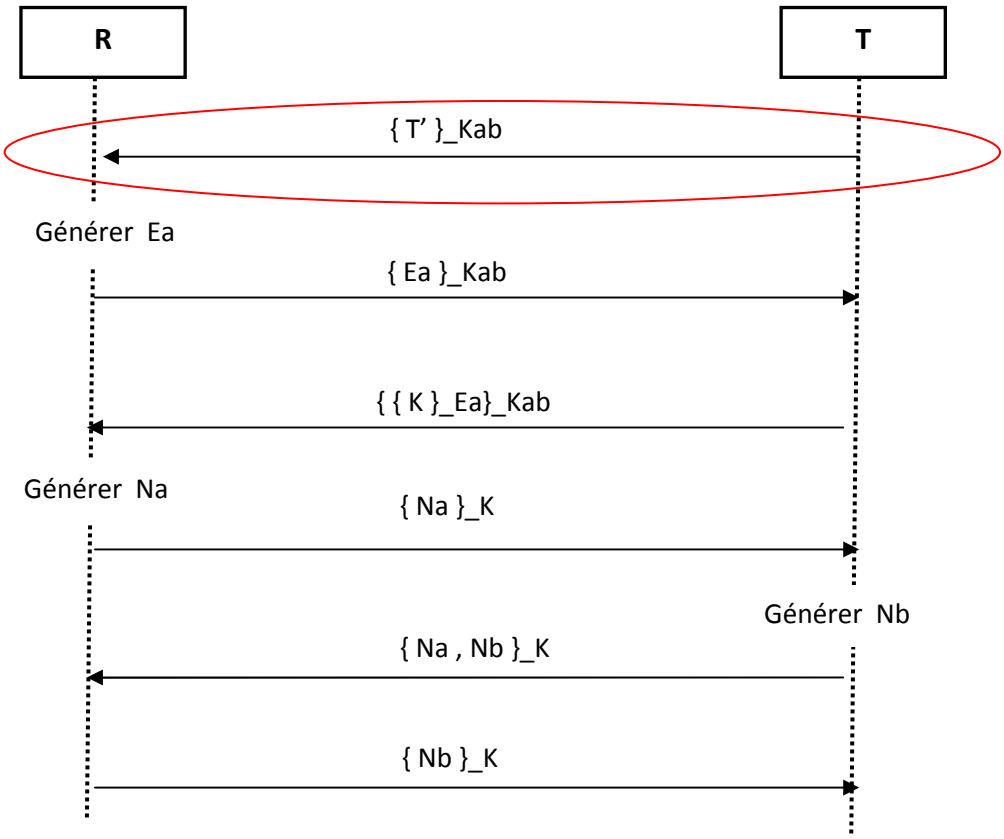
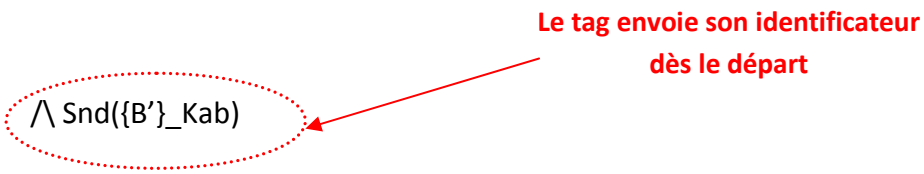


Figure IV.5: le protocole EKE amélioré

La partie modifiée dans la spécification HLPSL est les rôles de base :

```
role alice (A,B: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by A
def=

  local State   : nat,
      Ea, Na,Nb,K: text

 const sec_k1 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\  Rcv(start)  /\  Rcv({B'}_Kab)  =|>   State' := 1
                  /\ Ea' := new()
                  /\ Snd({Ea'}_Kab)

   2. State = 1  /\ Rcv({{K'}_Ea}_Kab) =|>  State' := 2
              /\ Na' := new()
              /\ Snd({Na'}_K')
              /\ secret(K',sec_k1,{A,B})
              /\ witness(A,B,na,Na')

   3. State = 2 /\ Rcv({Na.Nb'}_K)  =|>  State' := 3
             /\ Snd({Nb'}_K)
             /\ request(A,B,nb,Nb')

end role


role bob (B,A: agent,
            Kab: symmetric_key,
            Snd,Rcv: channel(dy))
played_by B
def=

  local State   : nat,
       Ea,Na,Nb ,K: text

 const sec_k2 : protocol_id

  init  State := 0

  transition

   1.   State = 0 /\ Rcv({Ea'}_Kab) =|> State' := 1
```

Le lecteur reçoit l'identificateur du tag dés le départ



/\ Snd({B'}_Kab)

/\ K' := new()
/\ Snd({{K'}_Ea'}_Kab)
/\ secret(K',sec_k2,{A,B})

2. State = 1 /\ Rcv({Na'}_K) =|> State' := 2
/\ Nb' := new()
/\ Snd({Na'.Nb'}_K)
/\ witness(B,A,nb,Nb')

3. State = 2 /\ Rcv({Nb}_K)=|> State' := 3
/\ request(B,A,na,Na)

end role

Après la vérification de protocole EKE amélioré par l'outil OFMC d' AVISPA, le résultat est comme suit:

```
% OFMC

% Version of 2006/02/13

SUMMARY

  SAFE

DETAILS

  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL

  C:\progra~1\SPAN\testsuite\results\EKE MIDM corriger.if

GOAL

  as_specified

BACKEND

  OFMC

COMMENTS

STATISTICS

  parseTime: 0.00s

  searchTime: 0.01s

  visitedNodes: 1 nodes
```

IV.7 Comparaison de la sécurité :

Le Tableau suivant illustre une comparaison de la sécurité avec les protocoles vérifiés :

protocole	EKE	EKE amélioré	Le protocole proposé
Résistance à l'attaque par rejeu	oui	oui	oui
Résistance à l'attaque MITM	non	oui	oui

Tableau IV.1: Analyse de la sécurité.

Complexité du tag :

La complexité de toutes les implémentations de primitives cryptographiques exigées devrait être la plus faible possible , d'où le coût du tag, i.e. complexité du tag, sera aussi faible. Le tableau suivant illustre les primitives cryptographiques exigées dans le tag selon le protocole d'authentification :

Protocole	**EKE**	**Le protocole proposé**
Fonction de hachage		x
PRNG	x	x
Cryptage symétrique	x	

Tableau IV.2: complexité du tag.

Dans notre travail, le protocole d'authentification EKE exige le chiffrement symétrique, généralement les algorithmes symétrique implémentés dans les systèmes RFID sont les algorithmes de catégorie chiffrement par bloque (block cipher) comme algorithme AES ,sa mise en œuvre a été réalisée en utilisant environ 3400 portes logiques sous forme de blocs de taille 128 bits (avec une fréquence d'horloge maximale estimée à 80MHz et la consommation d'énergie 8.2 µA dans 100kHz).

Le protocole proposé exige une fonction de hachage qui est une primitive cryptographique. Yüksel [13] a présenté l'implémentation de faible coût des fonctions de hachage, en utilisant seulement 1700 portes logiques sous forme de blocs de taille 64 bits (avec une fréquence d'horloge maximale estimée à 100 MHz).

Les deux protocoles étudiés auparavant exigent un générateur des nombres pseudo-aléatoire (PRNG) qui sert à générer des nonces. La mise en œuvre de ce générateur peut appeler une fonction de hachage .

Donc, en ce qui concerne la complexité, le tag du protocole proposé est de coût bas par rapport au tag du protocole EKE.

IV.8 Conclusion :

La vérification formelle est une méthode qui permet la description mathématique d'un protocole pour afin de vérifier la fiabilité des protocoles cryptographiques.

Dans ce chapitre nous avons présenté quelques notions de base de la vérification ainsi que ses objectifs, puis on a présente la vérification formelle : ses caractéristiques et ses méthodes ainsi, un ensemble d'outils qui permet d'assurer cette vérification, nous avons basé sur l'outil AVISPA, et le langage HLPSL .
Ensuite nous avons vérifié formellement le protocole cryptographique propose en utilisant les outils AVISPA & SPAN. Nous avons montré l'importance de cette vérification pour assurer les propriétés de confidentialité et d'authentification dans les systèmes RFID. Suite à nos vérifications, nous avons constaté que notre protocole est sûr.

Références bibliographiques

[1]: Nicolas Seriot. << Les systèmes d'identification radio (RFID) fonctionnement, applications et dangers >>. Article, IL-2005B, 13 janvier 2005.
[2]: Klaus finkenzeller. << RFIF handbook fundamentals and applications in contactless smart cards, radio frequency identification and near field communication >>. Book, third edition page 2-7, 2010.

3]: Jeremy Landt. << The history of rfid. Potentials >>, IEEE, 24(4):8-11, Oct 2005.

[6] : Belrepayre Sylvain. << La technologie RFID et le protocole Modbus >>. Exposé, université PARIS-EST, 01/02/2013

[7]: BRUN-MUROL Pierre. << Vers une méthodologie normalisée d'évaluation des solutions RFID en application de sécurité >>. Mémoire présenté en vue de l'obtention du diplôme de maitrise sciences appliquées (génie informatique), Ecole polytechnique de Montréal, avril 2013.
[8]: Harvey Lehpamer. << RFID Design Principles >>. ARTECH HOUSE, Boston, 2008.

12]: BACHOTI Youssef, BELHAJ SENDAGUE Bassim, RODRIGUES OLIVEIRA et Joao Gabriel. <<PROJET RFID >>**.** Projet de fin d'étude. Paris Tech : Institue des sciences et technologies, 25 janvier 2011.

[13]: Stevan Preradovic and Nemai Chandra Karmakar. <<Rfid transponders>>. In International Conference on Electrical and Computer Engineering, pages 96–99, Dec 2006.

[14] : Samuel Fosso Wamba. << Les impacts de la technologie RFID et du réseau EPC sur la gestion de la chaine d'approvisionnement >>. Thèse de doctorat, université de Montréal, septembre 2009.

[15]: Dat Son Nguyen. << Développement des capteurs sans fil basés sur les tags RFID uhf passifs pour la détection de la qualité des aliments >>. Thèse de doctorat, Soutenue a Université de Grenoble, Septembre 2013.

[16]: Pierre-Henri THEVENON. << Sécurisation de la couche physique des communications sans contact de type RFID et NFC>>. Thèse de doctorat, université de Grenoble, 10 novembre 2011.

[17] : Anthoy Ghiotto. << Conception d'antennes de tags RFID UHF, Application a la réalisation par jet de la matière >>. Thèse de doctorat, université de Grenoble, novembre 2008.

[18]: Sanjay Sarma. << A history of the EPC in RFID: Application, Security and Privacy >>. Chapitre extrait de: RFID: applications, security, and privacy, pages 37–55. Addison-Wesley, Boston, London, Dec 2005.

[19]: Dat Son Nguyen. << Développement des capteurs sans fil basés sur les tags RFID uhf passifs pour la détection de la qualité des aliments >>. Thèse de doctorat, Soutenue a Université de Grenoble, Septembre 2013.

[20]: Mandeep Kaur, Manjeet Sandhu, Neeraj Mohan and Parvinder S. Sandhu. << Rfid technology principles, advantages, limitations and its Applications >>. International Journal of

[21]: Marie Lise Flottes, LIRMM, Giorgio Di Natale, LIRMM, Guy Gogniat. << Sécurité Des Systèmes Embarqué >>. Article, Lab-STICC, 2011.

[22]: HAMADOU Sardaouna**.** << Analyse formelle des protocoles cryptographiques et flux d'information admissible >>. Thèse de doctorat, université de Montréal École Polytechnique, mars 2008.

[23]: Bernard COUSIN, << Sécurité des réseaux informatiques>>. Support de cours, université de Rennes 1.

[24]: Philippe MARTIN et Refik MOLVA. << Analyse de la sécurité >>. La source de la section : Analyse de la sécurité du document SP 1.2. (Étude prospective des besoins dans un réseau RFID communautaire) du projet PAC-ID GD, http://www.eurecom.fr/~martinph/PACID/SP1_2secur.html.

[25]: Jérémy Briffaut. << Formalisation et garantie de propriétés de sécurité système : application à la détection d'intrusions >>. Thèse de doctorat, universite d'Orleans déparement informatique, 13 décembre 2007.

[26]: Mathieu Baudet. << Sécurité des protocoles cryptographiques : aspects logiques et calculatoires >>**.** Thèse de doctorat, université de Cachan, 16 janvier 2007.

[27]: Heinrich Hordegen. << Vérification des protocoles cryptographiques : Comparaison des modèles symboliques avec une application des résultats — Etude des protocoles récursifs >>. Thèse de doctorat, université de Nancy, 29 Novembre 2007.

[29]: https://ssi.ac-strasbourg.fr/

[30]: G.Avoine and P.Oechslin. <<RFID traceability: A multilayer problem>>. In A.Patrick and M.Yung, editors, Financial Cryptography -FC'05, pages 125-140. Springer-Verlag, 2005.

[31]: A.Juels, S.Garfinkel, and R.Pappu. <<RFID privacy: An overview of problems and proposed solutions>>. IEEE Security and Privacy, 3(3):34-43, May/June 2005.

[32]: A.Karygiannis, T.Phillips, A.Tsibertzopoulos. <<RFID security: A taxonomy of risk>>. In Proceedings of the 1st international conference on communications and networking in China (ChinaCom'06) (pages 1-7). Beijing: IEEE. 2006.

[33]: Ding Zhen-hua, Li Jin-tao, and Feng Bo. << A Taxonomy Model of RFID Security Threats>>. Communication Technology, ICCT, pages 765-768, November 2008.

[34]: L. Mirowski, J. Hartnett, and R. Williams. <<An RFID Attacker Behavior Taxonomy>>. IEEE Pervasive Computing, 8(4):79-84, pages 79-84, October/December. 2009.

[35]: A.Mitrokotsa,MR. Rieback,AS. Tanenbaum. <<Classifying RFID attacks and Defenses>>. Special Issue on Advances in RFID Technology, Information Systems Frontiers, Springer Science and Business Media, LLC 2009. doi: 10.1007/s10796-009-9210-z., July 2009.

[36]: A. Mitrokotsa, M. Beye, P. Peris-Lopez, chapter: <<Security Primitive Classification of RFID Attacks>>. In Book: Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Eds. D. Ranasinghe, M. Sheng, S. Zeadally. Springer-Verlag. 2011.

[37]: Hong Li, YongHui Chen, ZhangQing He. <<The Survey of RFID Attacks and Defenses>>. Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, vol., no., pages 1-4, 21-23, September 2012.

[38]: Torstein HAVER. <<Security and Privacy in RFID Applications>>. Thèse de Master, Norwegian University of Science and Technology Department of Telematics, juin 2006.

[39]: Coulton, P., Rashid, O., and Bamford, W. << Experiencing 'touch' in mobile mixed reality games, Proceedings of The Fourth Annual International Conference in Computer Game Design and Technology>>, Liverpool, Novembre 2006.

[40]: Tuyls, P. & Batina, L. <<RFID-tags for anti-counterfeiting>>, In D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006, Vol. 3860 of LNCS, pp.115-131, Springer-Verlag, ISSN: 0302-9743, San Jose, CA, USA.

[41]: Mojtaba Alizadeh, Mazdak Zamani, Ali Rafiei Shahemabadi,JafarShayan and Ahmad Azarnik. <<A Survey on Attacks in RFID Networks>>. Open International Journal of Informatics (OIJI), VOL1, Universiti of Malaysia, 2012

[43]: Monty L. << Vulnérabilités des RFID >>, Article, 2010.

[44]: Stefan Brands and David Chaum. <<Distance-bounding protocols>>. Advances in Cryptology EUROCRYPT '93, Springer-Verlag LNCS 765, pp 344–359, May 1993.

[45]: Léonard Gross. << Sécurité RFID et préservation de la sphère privée >>. Mémoire de fin d'étude, Institute d'Information et de Communication, Suisse, Janvier 2007.

[**46]**: Rieback, M. R.; Crispo, B. & Tanenbaum, A. S. <<Is your cat infected with a computer virus? >>. Proc. of the 4th Annual IEEE International Conference on Pervasive Computing and Communication, pp. 169-179, ISBN: 0-7695-2518-0, 13-17 March 2006, Pisa, Italy.

[47]: Xavier Lemarteleur. <<Traçabilité contre vie privée : les RFIDs Ou l'immixtion des technologies dans la sphère personnelle>>. Mémoire de fin d'étude, université Paris II – Panthéon / Assas, octobre 2004.

[48]: Dirik Henrici. <<RFID security and privacy, concepts, protocols, and architectures>>. pages 45-55, 2008.

[49]: Mohammad Reza Sohizadeh Abyaneh. << security analysis of lightweight schemes for RF ID systems>>. Thèse de doctorat, université de Bergen norway, juin 2012.

[50]: Xu Zhuang · Yan Zhu & Chin-Chen Chang. << A New Ultralightweight RFID Protocol for Low-Cost Tags: R2AP>>. Wireless Pers Commun (2014) 79:1787–1802 DOI

10.1007/s11277-014-1958-x. Springer Science+Business Media New York 2014, 26 July 2014.

[51]: Ha, J., S.-J. Moon, J. M. G. Nieto and C. Boyd, <<Low-cost and strong-security RFID authentication protocol>>, in: EUC Workshops, 2007.

[52]: T. van Deursen and S. Radomirović. << Attacks on RFID Protocols>>. In Proc. 4th International Workshop on Security and Trust Management (STM'08), ENTCS. Elsevier, August 2009.

[53]: T. van Deursen and S. Radomirović. <<Security of RFID protocols- A case of study>>. In Proc. 4th International Workshop on Security and Trust Management (STM'08), ENTCS. Elsevier, Juin 2008.

[54]: Hung-Yu Chien and Chen-Wei Huang. <<A lightweight RFID protocol using substring>>. In Embedded and Ubiquitous Computing (EUC), 2007.

[55]: Lee et al. <<RFID mutual authentication scheme based on synchronized secret Information>>. In Symposium on Cryptography and Information Security, Hiroshima, Japan, January 2006.

[56]: Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. <<LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags>>. In *Proceedings of 2nd workshop RFID security*. Graz, Austria: Ecrypt. 2006.

[57] : Pascal Lafourcade. << Vérification de protocoles cryptographiques en présence de théories équationnelles>>. Thèse de doctorat, école normale supérieure de CACHAN Laboratoire Spécification et Vérification CNRS UMR 8643, lundi 25 septembre 2006.

[58]: Amina Cherif, Damien Sauveron and Malika Belkadi. << Overview on Formal Verification Methods for RFID Protocols >>. Article, université de Tizi Ouzou, Laboratoire LARI,Computer Science Department, Algeria, Université de Limoges, XLIM UMR CNRS 7252 – Mathematics and Computer Science Department, Limoges, France. 2016.

[59]: Hamdi NASSER, Michel FANGAYOUMANI et Simon DEBRAS. << Vérification formelle d'un protocole de sécurité à l'aide d'un outil d'analyse automatique des failles de sécurité >>. Rapport de projet, ENAC, 8juin 2011.
Computer Society.journal, 1997.