République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Réseaux, Mobilité et Systèmes Embarqués**

Présenté par
Malika MOHAMEDI
Samia IKERBANE

Thème

Vérification automatique d'un protocole de sécurité dans les systèmes RFIDs à base d'outils AVISPA & SPAN.

Mémoire soutenu publiquement le 29/ 06 / 2016 devant le jury composé de :

Président : Mr Mehammed Daoui
Encadreur : Mme Malika BELKADI
Examinateur : Mme Rachida AOUDJIT
Examinateur : Mme Rebiha Hadaoui

REMERCIEMENT

Au nom D'ALLAH le Miséricordieux ;

Nous voudrons adresser nos remerciements les plus sincères aux enseignants de département informatique (tizi ouzou) pour tout ce qu'ils nous ont apporté durant notre cursus ;

Nous tenons également à remercier sincèrement notre promotrice M^me^ M.Belkadi qui nous a proposé ce sujet, orienté, soutenu et conseillé tout au long de sa réalisation ;

Nous tenons à remercier aussi chaleureusement Melle A.Chérif pour son aide et encouragement tout au long de durée de ce mémoire ;

Nos sincères remerciements vont aussi au membre du jury pour le temps qu'ils ont passé à la lecture de ce mémoire ;

Pour finir, nos dernières pensées vont vers nos familles, et surtout à nos parents, pour les sacrifices et les moyens qu'ils ont mis à notre disposition afin de nous permettre de suivre nos études dans les meilleurs conditions ;

Malika ; Samia… ✍

Dédicaces

Je dédie ce modeste travail ;

A mes très chers parents pour leurs sacrifices et leurs

Encouragements que dieu les protègent ;

A la mémoire de ma grand-mère que dieu lui offre le paradis eternel ;

A celle avec qui j'ai partagé de bons moments : ma tente fatma ;

A ceux qui m'ont encouragé et aidé et qui seront fière de moi : ma grande famille que j'aime beaucoup ;

A ceux que j'aime et que je respecte : mes vraies amies ;

A mon binôme et sa famille ;

Malika

Dédicaces

Je dédie ce modeste travail ;

A mes très chers parents pour leurs sacrifices et leurs

Encouragements que dieu les protègent ;

A la mémoire de mes grands-parents que dieu les offre le paradis eternel;

A mes frères :Rabah, Lounes, Moumoh et Ali ;

A mes tantes et mes oncles ;

A ceux qui m'ont encouragé et aidé et qui seront fière de moi : ma grande famille que j'aime beaucoup ;

A ceux que j'aime et que je respecte : mes vraies amies ;

A mon binôme et sa famille ;

Samia

Résumé de mémoire

La RFID (Radio Frequancy IDentification) est une technologie d'identification qui suscite de plus en plus un grand intérêt d'utilisation dans divers domaines. Néanmoins, un des plus importants challenges de ces systèmes RFID est la sécurité. Pour garantir une bonne sécurité et protéger la vie privée des utilisateurs de ces systèmes, la cryptographie est utilisée. Comme les systèmes RFIDs sont caractérisés par leurs ressources limitées (mémoires, puissance de calcul), les protocoles cryptographiques destinés pour ces systèmes doivent utiliser des primitives légères ou ultra légères. Beaucoup de protocoles cryptographiques sont déjà proposés pour les systèmes RFID. Malheureusement, des vulnérabilités et des failles de sécurité ont été découvertes dans la plupart de ces protocoles, en dépit de la quantité considérable de temps et d'efforts requis pour la conception et la mise en œuvre de ces protocoles. La vérification de ces derniers avant leur utilisation devient un besoin cruciale.

Dans ce travail, nous nous sommes intéressés à la vérification formelle de protocoles cryptographiques à base de l'outil AVISPA&SPAN. Nous avons vérifié deux protocoles : Le premier (R^2AP) s'est révélé sans faille. Par contre, le deuxième (HMNB) est vulnérable vis-à-vis de deux attaques (attaque par rejeu et attaque main in-the-middle). A ce dernier, nous avons proposé une amélioration qui est prouvée résistante aux attaques précédentes à base d'AVISPA.

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

La liste des tableaux

Tableau I.1 : Comparaison entre les systèmes d'identification automatique 6

Tableau I.2 : Gammes de fréquences utilisées dans les systèmes RFID..... 10

Tableau I.3 : Norme ISO 18000 pour la standardisation des systèmes RFID..... 12

Tableau II.1 : les mesures de sécurité dans un système RFID 38

Tableau II.2 : Résumé des attaques sur les RFID 35

Tableau III.1 : Notations utilisées dans R^2AP..... 41

Tableau III.2 : Notations utilisées dans HMNB..... 45

Tableau III.3 : Mises à jour dans HMNB..... 47

Tableau IV.1 : Notations utilisées dans la spécification HLPSL de R^2AP 60

Tableau IV.2 : Notations utilisées dans la spécification HLPSL de HMNB 69

Tableau IV.3 : Mises à jours du protocole amélioré 74

Tableau IV.4 : Analyse de la sécurité 79

Tableau IV.5 : Analyse des performances..... 80

Liste des figures :

Figure I.1 : Un code a barres EAN (European Article Number)..... 4

Figure I.2 : Principe de fonctionnement d'un système RFID..... 7

Figure I.3 : Exemples de tags RFID..... 7

Figure I.4 : Les types de lecteur RFID..... 9

Figure I.5 : Code EPC version 96 bits..... 13

Figure I.6 : La RFID dans les documents d'identité..... 14

Figure I.7: Le dispositif de paiement Pidion BIP-1300 14

Figure I.8 : La RFID dans le domaine de la santé..... 15

Figure I.9 : La RFID dans le secteur d'agriculture 16

Figure I.10 : La RFID dans le secteur de la logistique..... 16

Figure II.1 : Authentification par RFID 23

Figure II.2 : Identification par RFID..... 23

Figure II.3 : Attaque skmming..... 25

Figure II.4: Attaque eaversdropping 26

Figure II.5 : champ magnétique maximum à 10 m du lecteur..... 29

Figure II.6: Cage de faraday bloquent les signaux dans la gamme de fréquences 10MHz-20GHz... 29

Figure II.7: Présentation d'une attaque relais sans fil proposé par G.Hancke 30

Figure II.8: Attaque main in- the-middele 32

Figure III.1: Les protocoles légers dans les systèmes RFID 40

Figure III.2: Diagramme de séquences du protocole R^2AP 43

Figure III.3: Diagramme de séquences du protocole HMNB 46

Figure IV.1 : Architecture logicielle d'AVISPA 56

Figure IV.2 : Structure d'une spécification HLPSL 60

Figure V.1 : Trace d'attaque MITM de HMNB (CL-Atse) 71

Figure V.2 : Diagramme de séquences du protocole amélioré..... 73

Figure V.3 : Diagramme de séquences illustre l'attaque de désynchronisation dans HMNB 77

Figure V.4 : Diagramme de séquences illustre la résistance à l'attaque de désynchronisation dans le protocole amélioré..... 78

Figure A.1 : Architecture du système AVISPA et SPAN..... 82

Figure A.2 : Interface graphique de vérification..... 83

Figure A.3: L'interface SPAN lors d'une exécution interactive (Protocol simulation)..... 85

Figure A.4 : L'interface SPAN lors d'une exécution interactive (Intruder simulation) 86

Figure A.5 : Présentation de l'interface Attack simulation..... 87

Table des matières

Introduction générale **1**

Chapitre I : Généralité sur les systèmes RFID

I.1 Introduction **3**

I.2 Les systèmes d'identification automatique **3**

I.2.1 Les codes à barres **3**

I.2.2 Reconnaissance optique de caractères **4**

I.2.3 Cartes à puce **4**

I.2.4 La technologie RFID **4**

I.3 Historique de la RFID **4**

I.4 La comparaison entre les systèmes d'identification automatique **5**

I.5 Les composants et principes de fonctionnement des systèmes RFID **6**

I.5.1 Tag RFID **7**

I.5.1.1 Types des tags RFID **8**

I.5.1.2 La puce **8**

I.5.2 Le lecteur **9**

I.5.2.1 Types des lecteurs RFID **9**

I.5.2.2 L'antenne RFID **9**

I.5.3 La base de données **10**

I.6 Les gammes des fréquences utilisées dans les systèmes RFID **10**

I.7 Les normes RFID. **11**

I.7.1 Quelques standards et normes d'ISO **11**

I.7.2 Le standard EPC **13**

I.8 Quelques exemples d'application de la technologie RFID dans la vie quotidienne **13**

I.8.1 La RFID dans les documents d'identité 13

I.8.2 La RFID comme outil de paiement 14

I.8.3 La RFID dans le secteur de la santé 15

I.8.4 La RFID pour les animaux domestiques 15

I.8.5 La RFID dans la logistique 16

I.9 Les avantages et inconvénients 17

I.9.1 Avantages 17

I.9.2 Inconvénients 17

I.10 Quelques défis de la technologie RFID 18

I.11 Conclusion 18

Chapitre II : La sécurité dans les systèmes RFID

II.1 Introduction 20

II.2 Propriétés de sécurité 20

II.3 Attaques de sécurité 21

II.4 Mécanismes de sécurité 21

II.5 RFID : Attaques et Sécurités 22

II.5.1 Classification des applications RFID selon les besoins de sécurité 22

II.5.2 Types d'attaques et contre-mesures dans les systèmes RFID 23

II.5.2.1 Menaces de la couche physique 24

II.5.1.2 Menaces de la couche Transport 31

II.5.1.3 Menaces de la couche application 33

II.5.1.4 Autres attaques 33

II.5.3 Résumé des attaques et contre-mesures 35

II.5.4 Respect de la vie privée **36**

II.5.5 S'attendre à l'inattendu **37**

II.6 Conclusion **38**

Chapitre III : Les protocoles cryptographiques dans les systèmes RFID

III.1 Introduction **39**

III.2 Les protocoles cryptographiques dans les systèmes RFID **39**

III.2.1 Les protocoles lightweight **39**

III.2.2 Les protocoles ultralightweight **40**

III.3 Présentation de quelques protocoles d'authentification dans les systèmes RFID **41**

III.3.1 Le protocole R^2AP (Reconstruction based RFID Authentication Protocol) **41**

III.3.1.1 Notations **41**

III.3.1.2 Description du protocole **42**

III.3.2 Le protocole HMNB **45**

III.3.2.1 Notations **45**

III.3.2.2 Description du protocole **46**

III.3.3 Le protocole de Chien et Huang **47**

III.3.4 Le protocole de Lee et al **48**

III.3.5 Le protocole LMAP(Lightweight Mutual Authentication Protocol) **48**

III.4 Conclusion **48**

Chapitre IV : La vérification formelle et l'outil AVISPA

IV.1 Introduction **49**

IV.2 Les notions de base de la vérification **49**

IV.2.1 Les protocoles **49**

IV.2.2 Les agents **49**

IV.2.3 L'attaque **50**
IV.2.4 Secret **50**
IV.3 Les objectifs de la vérification **50**
IV.3.1 Trouver une attaque **50**
IV.3.2 La sûreté d'un protocole **50**
IV.4 Les modèles de vérification **50**
IV.4.1 La vérification calculatoire **51**
IV.4.2 La vérification formelle **51**
IV.5 Les méthodes de vérification formelle **52**
IV.5.1 La méthode du raisonnement logique **52**
IV.5.2 L'approche par preuve de théorèmes **53**
IV.5.3 Les méthodes d'exploration de l'état **53**
IV.6 Le modèle d'intrus de Dolev-Yao **53**
IV.6.1 Un réseau idéal **53**
IV.6.2 Abstraction des messages échangés **53**
IV.6.3 Chiffrement parfait **54**
IV.6.4 Capacités de l'intrus **54**
IV.7 Les outils de vérification formelle **54**
IV.7.1 L'outil AVISPA **54**
IV.7.2 L'outil ProVerif **54**
IV.7.3 L'outil Hermes **55**
IV.7.4 L'outil Casper **55**
IV.8 Présentation de l'outil AVISPA **55**
IV.8.1 Architecture logicielle de l'outil AVISPA (back-end AVISPA) **55**
IV.8.2 La démarche AVISPA **57**
IV.8.3 L'outil graphique SPAN **58**
IV.8.4 Les étapes de vérification **58**

IV.9 Présentation du langage de spécification des protocoles HLPSL **59**
IV.9.1 Structure d'une spécification HLPSL **59**
IV.9.1.1 Définition des rôles **60**
IV.9.1.2 Déclaration des propriétés de sûreté **61**
IV.9.1.3 Instanciation d'un rôle **61**
IV.9.1.3 Structure d'une spécification HLPSL **62**
IV.9.2 Caractéristiques de HLPSL **63**
Conclusion **64**

Chapitre V : Vérification formelle d'un protocole RFID à l'aide d'AVISPA & SPAN

V.1 Introduction **65**
V.2 Les vérification sur AVISPA & SPAN **65**
V.3 La vérification formelle du protocole R^2AP **65**
V.4 Vérification formelle du protocole HMNB **68**
V.5 Proposition d'une amélioration du protocole HMNB **71**
V.5.1 Description du protocole **72**
V.5.2 Analyse de ce protocole **74**
V.5.2.1 Vérification automatique **74**
V.5.2.2 Autre analyses **76**
V.5.3 Comparaison de la sécurité **79**
V.5.4 Analyse de la performance **79**
V.6 Conclusion **80**

Conclusion générale **81**
Annexe A **82**
Bibliographie **90**

Introduction générale

Les progrès dans le domaine de l'informatique et de l'électronique ont permis de donner corps à une idée assez ancienne, celle d'identification par radiofréquence qui est déjà utilisée lors de la deuxième guerre mondiale.

L'identification par radiofréquence ou RFID (pour Radio Frequency Identification en anglais) est une technologie sans fil qui permet de désigner et tracer des objets. Elle est de plus en plus utilisée dans de nombreuses applications et mises en œuvre dans divers secteurs, mais malgré le large essor de cette technologie elle rencontre cependant des obstacles qui nécessite la mise en place d'un cadre pour les politiques destinées à accroître les effets bénéfiques de cette technologie au profit des entreprises et des consommateurs tout en prenant en compte les questions de sécurité et de protection de la vie privée.

La sécurité de l'information est devenue une tâche primordiale, elle a divers objectifs, elle doit protégée les informations contre l'accès, l'utilisation, la destruction ou la modification non autorisée. La protection doit garantir l'authenticité, la confidentialité, l'intégrité des informations, selon les exigences de la situation particulière.

Parmi les différents aspects de la sécurité informatique utilisés pour garantir un certain niveau de sécurité : la cryptologie et l'étude des protocoles cryptographiques qui ont pris une ampleur considérable. La cryptologie est l'ensemble des moyens permettant de transmettre des messages d'une manière confidentiels uniquement compréhensible par le vrai destinataire.

Les protocoles de communication chiffrent, grâce à des codes secrets, leurs échanges de messages. 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. La découverte de telles attaques entraine une perte économique considérable qui oblige alors à retirer momentanément le système, afin de le corriger ou le remplacer. Il est donc indispensable de vérifier automatiquement la sécurité des protocoles cryptographiques avant leurs mises en service, afin de s'assurer qu'ils ne possèdent pas de failles. La sécurité de ces protocoles n'est pas uniquement garantie par l'usage d'une méthode de chiffrement cryptographique mais aussi par une vérification automatique du protocole lui-même.

Les tâches de vérification des propriétés de sécurité servent à prouver que les protocoles sont fiables. Vu que l'informatique est basée sur les mathématiques, les preuves du fonctionnement des programmes sont présentées par des formules mathématiques. Cette formalisation des processus, par des formules et des algorithmes mathématiques, nous amène dans un domaine que nous appelons les méthodes formelles, sur lesquelles se basent plusieurs outils de vérification automatique comme AVISPA.

Notre mémoire est structuré comme suit :

Chapitre I : Généralité sur les systèmes RFID.

Dans ce chapitre nous avons défini la RFID, ses composants, son système de fonctionnement, ainsi que les gammes de fréquences utilisées pour cette technologie, quelques domaines d'application, avantages et inconvénients et quelques défis de la RFID.

Chapitre II : La sécurité dans les systèmes RFID.

Dans ce chapitre nous avons présenté les propriétés, les attaques et mécanismes de sécurité, nous avons fait une classification des applications RFID selon les besoins de sécurité ensuite on a défini les types d'attaques et contre-mesures dans les systèmes RFID.

Chapitre III : Les protocoles cryptographiques dans les systèmes RFID.

Dans ce chapitre nous avons définie les types des protocoles cryptographiques dans les systèmes RFID, et nous avons données une présentation de quelques protocoles d'authentification de différents types.

Chapitre IV : La vérification formelle et l'outil AVISPA.

Dans ce chapitre nous avons défini quelques notions relatives à la vérification, les modèles de vérification en se basant sur la vérification formelle ses caractéristiques, ses méthodes, ainsi le modèle d'adversaire de Dolev et Yao utilisé dans cette vérification. Ensuite on a présenté quelques outils de vérification automatique en se focalisant sur la l'outil AVISPA qui est utilisé dans notre projet.

Chapitre V : Vérification formelle d'un protocole RFID à l'aide d'AVISPA.

Dans ce chapitre nous avons donné les étapes de vérification sur SPAN, Les protocoles d'authentification des RFID, ainsi l'étude des deux protocoles R2AP et HMNB: leur description et leurs spécification HLPSL et l'amélioration de HMNB.

CHAPITRE I : Généralités sur les systèmes RFID

I.1 Introduction

Parmi les systèmes qui ont été développés rapidement au cours de ces dernières années, ceux d'identification automatique (Auto-ID) qui fournissent des informations sur des personnes, des animaux, des biens et des produits en transit, ils sont devenus très populaires dans de nombreux domaines : les services d'industrie, les achats, la logistique, la médecine…etc.

La technologie RFID (pour Radio Frequency IDentification) est l'un des systèmes d'identification automatique qui a déjà été utilisé dans la deuxième guerre mondiale.

Dans ce chapitre nous allons donner un aperçu sur les systèmes d'identification automatique, une comparaison entre eux en mettant en évidence les puissances et les faiblesses des RFID, ensuite on passe à la présentation de généralités sur les systèmes RFID en décrivant leurs composants, leur fonctionnement, les gammes de fréquences utilisées dans ces systèmes, leurs domaines d'application, et nous terminons par les avantages, les inconvénients de cette technologie.

I.2 Les systèmes d'identification automatiques

La RFID fait partie du groupe des technologies d'identification automatique, au même titre que les codes à barres, la reconnaissance optique de caractères, les cartes à puce et autres technologies apparentées.

I.2.1 Les codes à barres

Il existe plusieurs technologies possibles pour réaliser l'identification d'objets ou de personnes. La plus connue et la plus répandue est probablement la technologie Code à Barres dont le premier principe a été introduit en 1948 par Bernard Silver [1].

Un code à barres représente la codification d'une information, variant selon l'algorithme de codage, il est représenté par une séquence de barres vides et de barres pleines, larges ou étroites, disposées parallèlement. La séquence peut être interprétée numériquement ou alpha numériquement. Elle est lue par balayage optique, c'est-à-dire lors de passage d'une source lumineuse sur les barres sombres et claires d'un code à barres l'intensité lumineuse varie, celle-ci captée par un capture photosensible, est amplifiée, filtrée et digitalisée pour être convertie en information numérique qu'un décodeur retranscrit en caractères ASCII directement exploitable par un système informatique [2].

Le code à barres le plus courant est le code EAN (European Article Number) illustré par la figure 1, créé pour répondre aux besoins de l'industrie alimentaire en 1976. Le code EAN est une évolution de l'UPC (Universal Product Code) américain, introduit aux Etats-Unis dès 1973. L'EAN est composé de 13 chiffres : l'identifiant de pays, l'identifiant de a société, le numéro de l'objet chez le fabricant, et un numéro de contrôle [1] [2].

Figure I.1 : Un code à barres EAN (European Article Number)

I.2.2 Reconnaissance optique de caractères

La reconnaissance optique de caractères (Optical Character Recognition, OCR) est utilisée depuis les années 1960. Elle fonctionne avec des polices de caractères conçues pour être lisibles aussi bien par les hommes que par les machines.

L'avantage le plus important des systèmes OCR est la haute densité de l'information et la possibilité de lire des données visuellement en cas d'urgence, ou tout simplement pour les vérifier, Cependant ces systèmes ont échoué à être applicable universellement en raison de leur prix élevé et les lecteurs compliqués qu'ils exigent, en comparaison avec d'autres méthodes d'identification [1] [2].

I.2.3 Cartes à puce

Une carte à puce est un système de stockage de données électroniques, éventuellement avec une capacité de calcul supplémentaire (carte à microprocesseur). Le contact entre la surface de la carte à puce et le lecteur établit une connexion, ce qu'alimente en énergie cette carte. Le transfert de données entre le lecteur et la carte a lieu à l'aide d'une interface série bidirectionnelle. Il est possible de faire la différence entre deux types de base de cartes à puce en fonction de leur fonctionnalité interne: la carte mémoire et la carte à microprocesseur [2].

I.2.4 La technologie RFID

L'identification par radiofréquence ou RFID est une technologie sans fil qui est utilisée pour identifier, suivre et tracer les éléments (objets, animaux et humains). Le concept de RFID a déjà été utilisé dans la deuxième guerre mondiale par l'Armée de l'Air britannique pour distinguer les avions alliés des avions ennemis [3] [4] [5].

Le système RFID se compose principalement de tags (étiquettes ou transpondeurs) et de lecteurs.

.

I.3 Historique de la RFID

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

- Le principe de la RFID est utilisé lors de la Seconde Guerre Mondiale 1940 pour identifier les appareils en vol (IFF : Identify Friendly or Foe), les alliés mettaient dans leurs avions des transpondeurs, le lecteur au sol agissait comme un radar qui interrogeait à distance les

avions en envoyant des ondes radio, par réflexion d'ondes, le transpondeur, s'il était présent dans l'avion, permettait aux équipes au sol d'identifier si l'engin était allié ou ennemi.

- Durant les années 1970, les systèmes RFID restèrent une technologie protégée à usage militaire supportée par les états pour la sécurité de sites sensibles notamment dans le secteur du nucléaire.

- L'année 1980 est marquée par l'apparition des tags passifs grâce aux avancées technologiques, plusieurs sociétés européennes et américaines se mettent à les fabriquer. Une des premières applications commerciales est l'identification de bétail en Europe.

- Dans les années 1990**,** le début de la normalisation pour une interopérabilité des équipements RFID : tags et lecteurs de manière générale. De plus, la miniaturisation du système RFID permette sont intégration dans une seule puce électrique par l'IBM[1].

- Fondation de l'Auto-ID center : centre de recherche spécialisé en identification automatique (entre autre RFID) par le MIT[2] en 1999.

- En 2004, l'Auto-ID du MIT devient « EPC global » qui est une organisation chargée de promouvoir la norme EPC (Electronic Product Code) sorte de super code à barres stocké dans un tag RFID, cette norme est élaborée par les universitaires et adoptée par l'industrie.

- A partir du 2005**,** les technologies RFID sont quasiment répandues dans tous les secteurs (santé, aéronautique, transport…).

- En 2009**,** la Création de centre national de référence RFID(CNRFID) qui a pour objectif d'encourager le déploiement de solution RFID et de développer les partenariats entre les offreurs de solution, les utilisateurs, les laboratoires de recherche et les institutions.

I.4 La comparaison entre les systèmes d'identification automatique

Une comparaison entre les systèmes d'identification décrite dans le tableau I.1 selon quelques critères met en évidence les forces et les faiblesses de la RFID par rapport aux autres systèmes [2]:

[1] IBM **(**pour International Business Machines Corporation) est une société multinationale américaine présente dans les domaines du matériel informatique, du logiciel et des services informatiques.

[2] MIT (pour Massachusette Institue of Technology) est un institut de recherche et une université américaine, spécialisée dans les domaines de la science et de la technologie.

Les paramètres du système	**Code à barres**	**OCR**	**Carte à puce**	**Système RFID**
La quantité de données (octets)	1-100	1-100	16 - 64k	16 - 64k
La densité de données	faible	faible	très haute	très haute
Lisibilité par machines	bonne	bonne	bonne	bonne
Lisibilité par des personnes	limitée	simple	impossible	impossible
Influence de la saleté et l'humidité	très grande	très grande	possible (contacte)	pas d'influence
Coût d'achat de lecteur électronique	très faible	moyen	faible	moyen
Vitesse de lecture	faible	faible	faible	Très vite
Distance maximale entre le support de données et le lecteur	0 - 50cm	<1cm	Contact direct	0 – 5m

Tableau I.1 : Comparaison entre les systèmes d'identification automatique. [2]

I.5 Les composants et principes de fonctionnement des systèmes RFID

Un système RFID se compose principalement d'un lecteur ou plusieurs, d'un tag ou plusieurs, et d. La figure I.2 décrit le schéma général d'un système RFID [7] [8].

La RFID utilise le rayonnement radio fréquence pour la communication entre le lecteur que l'on appelle aussi interrogateur ou station de base, et l'étiquette également nommée transpondeur ou tag, incorporé dans l'objet à identifier.

Le principe de fonctionnement est présenté sur la Figure I.2 : le lecteur émet un champ électromagnétique autour de lui. Le, ou les, tags qui se situent dans le champ d'émission reçoivent, par l'intermédiaire d'une antenne, le champ électromagnétique qui va les activés. Ces derniers vont alors renvoyer un signal et un dialogue va ainsi être établi entre les tags et le lecteur suivant un protocole de communication prédéfini. Des données peuvent ainsi être échangées, notamment les identifiants uniques de chaque transpondeur qui permettront l'identification [9].

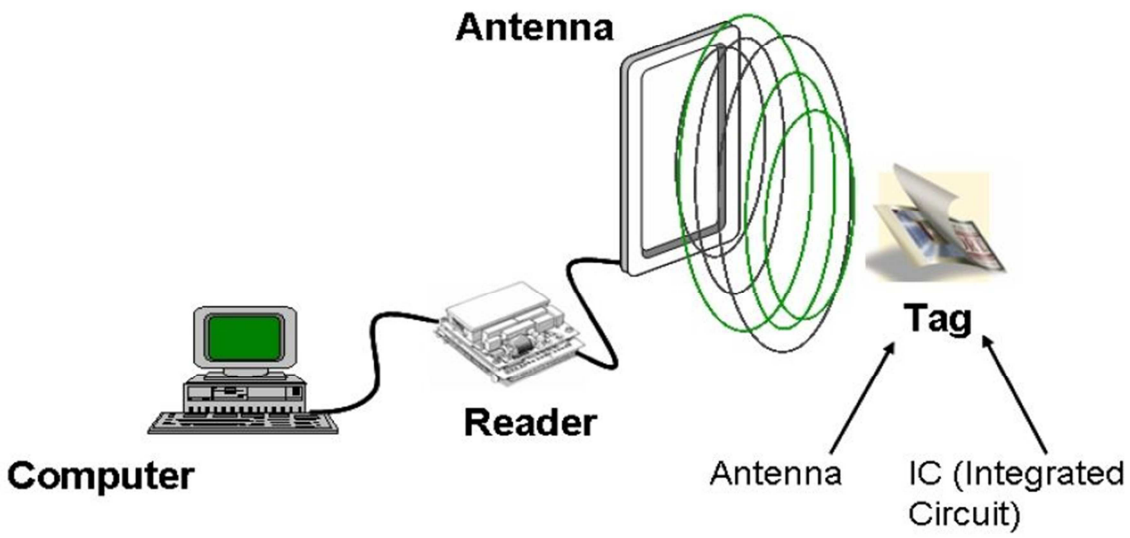


Figure I.2: Principe de fonctionnement d'un système RFID. RFID. [10]

I.5.1 Tag RFID

Appelée aussi étiquette ou transpondeur (pour TRANSmitter/rePONDER) comprend une puce (chip) dotée d'une mémoire qui mémorise les données numériques, reliée à une antenne bobinée qui transmet les données. Le plus souvent, le tag est collé sur un film en plastique ou moulé dans une carte. Des exemples de tag sont donnés dans la figure I.3

Le format des données inscrites sur l'étiquette est standardisé à l'initiative d'EPC global (Electronic Product Code).

La taille des tags RFID est éminemment variable selon les applications, elle dépend de la capacité de traitement de processeur, de la distance entre le lecteur et les étiquettes, de l'environnement de travail… [11].

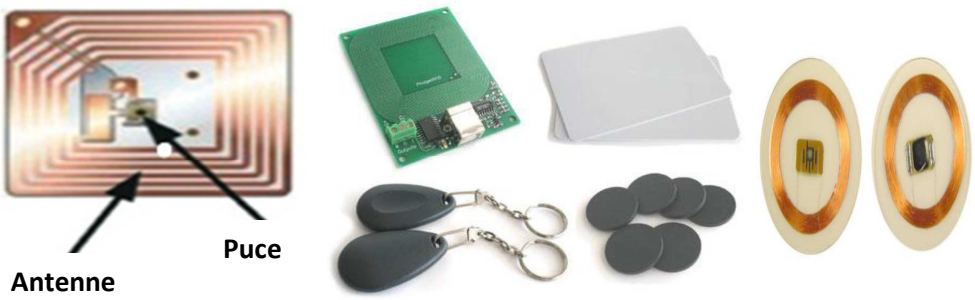


Figure I.3 : Exemples de tag RFID

I.5.1.1 Types des tags RFID

Il existe plusieurs classifications de tags, mais la plus importante est basée sur leur alimentation: passive, active ou semi-passive [6] [12] [13]:

- ***Tags passifs :*** ils ne possèdent pas leur propre alimentation en énergie, leur activation dépend de l'effet électromagnétique de réception d'un signal émis par le lecteur. C'est ce courant qui leur permet d'alimenter leurs microcircuits, ils modulent l'onde émise par lecteur et retourner ce signal sur lequel l'information des tags est superposé cette technique est dite la rétro-modulation. Ils sont peu coûteux à produire et sont généralement réservés à des productions en volume. Ce sont eux que l'on trouve plus particulièrement dans la logistique et le transport, selon leur capacité de transmettre à distance et au travers de substances différentes (air, eau, métal). La distance de lecture est inférieure à un mètre.

- ***Tags semi-passifs :*** Ils intègrent tout de même une source d'énergie interne (batterie), cela permet d'augmenter la distance de communication et la rapidité en lecture et en transmission que les tags passifs mais fait également augmenter leur coût et leur taille. La communication entre le lecteur et le tag fait appel aux techniques de retro-modulation et n'utilise pas la batterie interne qui sert à alimenter l'électronique intégré sur le tag.

- ***Tags actifs :*** Les tags actifs disposent d'un émetteur à haute fréquence, le tag retourne les informations en produisant lui même une onde électromagnétique qui génère un signal de sortie. Cette source d'alimentation supplémentaire procure aux puces actives plusieurs avantages par rapport aux passives. 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, sont plus gros, et plus chers à produire.

I.5.1.2 La puce

La puce RFID est élément important de tag qui sert à stocker et mémoriser les données, il existe trois types modes de fonctionnement de tags RFID selon la programmation de la puce RFID [11]:

- ***Puce à lecture seule (read only : RO) :*** Ne peut être programmée qu'une seule fois. Les données peuvent être enregistrées dans la mémoire de la puce lors de la fabrication. Après cette écriture, on ne peut plus changer les données sur la puce, c'est le mode de fonctionnement le plus simple du transpondeur et qui sert principalement pour les problèmes traitant seulement d'identification.

- ***Puce à écriture unique/lecture multiple (Write Once Read Multiple : WORM) :*** Ce mode de fonctionnement est similaire à la lecture seule, sauf que cette fois ci, le tag livré par le fabriquant est vierge, il ne contient aucune information. C'est l'acheteur du tag qui va pouvoir venir écrire des informations dessus. Les informations ne peuvent être écrites qu'une seule fois, ensuite le tag se comporte comme un tag en lecture seule.

- ***Puce à lecture/écriture multiples :*** 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. Ce type de tag est très utile lorsque l'on vient écrire des informations à différents moments d'un processus ou bien lorsque l'on souhaite réutiliser les tags avec de nouvelles informations.

I.5.2 Le lecteur

Appelé aussi interrogateur ou station de base, est un dispositif électronique couplé à une ou plusieurs antennes. Il est composé de : module radio (émetteur et récepteur) qui est un dispositif électronique qui reçoit et décode les informations envoyées par le tag et les envoie au dispositif de collecte des données, une source d'énergie qui alimente l'antenne radio qui permet la communication avec le tag. Le lecteur émet des ondes radio par l'intermédiaire de l'antenne, puis écoute les réponses des tags qui se trouvent dans son champ de lecture. La plupart des lecteurs fournissent une interface supplémentaire de type RS 232 pour transférer les données reçues à d'autres systèmes (PC, robots, etc.) [1] [11] [14].

I.5.2.1 Types des lecteurs RFID

Les lecteurs RFID ont des formes variées selon l'objectif d'utilisation pour lequel sont crées, il existe trois types qui sont illustrés dans la figure I.4 [14]:

- ***Lecteur mobile :*** qui est en général montés sur des chariots, offrant ainsi une mobilité et une flexibilité accrues dans les applications de type gestion d'entrepôts.

- ***Lecteur portatif :*** Dans ce type de lecteurs, les objets n'ont pas besoin d'être transporté, c'est le lecteur qui se déplace, qui est en général utilisé dans les applications de recherche et de localisation de produits dans un entrepôt et dont les antennes intégrées sont incorporées directement dans le dispositif.

- ***Lecteur fixe :*** il est installé de manière fixe et ne peut pas être donc transporté pour la lecture des puces à distance**.**

Figure I.4 : Les types de lecteur RFID.

I.5.2.2 L'antenne RFID

Est un élément essentiel pour le fonctionnement des systèmes RFID, intégré dans les tags et dans les lecteurs. Selon le type de lecture, le type d'étiquette, l'utilisation du système RFID, les conditions de lecture, il existe deux types principaux d'antennes [5]:

- ***Les antennes intégrées :*** elles sont intégrées au lecteur, elles sont utilisées pour les lectures à basse fréquence à portée limitée.

- ***Les antennes externes :*** elles sont plus puissantes, et s'avèrent donc utiles pour obtenir une plus grande portée.

I.5.3 La base de données

Les données acquises par les lecteurs de tag sont généralement transmis à un système base de données. Un certain type de logiciel de base de données contrôle ces informations reçues. La communication entre le lecteur et la base de données est considéré comme un canal sécurisé.

I.6 Les gammes des fréquences utilisées dans les systèmes RFID

Le choix de la fréquence dans un système RFID dépend de nombreux facteurs comme le type d'application, le type de produit, l'environnement d'opération, la distance de lecture….Ces systèmes doivent veiller a ne pas perturber le fonctionnement des autres systèmes, en ne peut utiliser que les plages de fréquences réservées aux applications ISM(Industriel-Scientifique-Médicale), le tableau I.2 présente les principales plages de fréquences utilisées par les systèmes RFID [16]:

Fréquences	Distance de lecture	Vitesse de transmission	Applications
Basses Fréquences 125-148 Khz	Jusqu' à 1.2m selon les normes 11784, 11785 et 14223.	lente	Suivi d'objets, identification d'animaux...
Hautes Fréquences 13,56 Mhz	ISO14443 : jusqu'à 30 cm ISO15693 et ISO18000 : jusqu'à 3 mètres	moyenne	Contrôle d'accès. Transport, traçabilité, porte monnaie électronique…
Très Hautes Fréquences 865 - 956 MHz	Grande distance de lecture jusqu'à 5 mètres selon la norme ISO18000.	rapide	Identification, logistique, gestion de stocks Multiples.
Ultra Hautes Fréquences 2,4 Ghz	Très grande distance de détection (>10 mètres) selon la norme ISO18000.	Très rapide	Droit de passage d'autoroute. Tag alimenté (actif).

Tableau I.2 : Gammes de fréquences utilisées dans les systèmes RFID. [16]

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 :

I.7.1 Quelques standards et 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 I.3 [17]:

Références	Fréquences concernées	Intitulé
18000-1	Vocabulaire et définitions	RFID pour la gestion d'objets-partie1 : Architecture de référence et définition des paramètres à normaliser (vocabulaire).
18000-2	<135 KHz	RFID pour la gestion d'objets-partie2 : Paramètres de communications d'une interface d'air en dessous de 135kHz.
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.
18000-4	2.45 GHz	RFID pour la gestion d'objets-partie4 : paramètres de communications d'une l'interface d'air à 2,45 GHz
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.
18000-6	960 MHz	RFID pour la gestion d'objets-partie6 : Paramètres de communications d'une interface radio entre 860 MHz et 960 MHz.
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.

Tableau I.3 : Norme ISO 18000 pour la standardisation des systèmes RFID. [17]

I.7.2 Le standard EPC

Parmi toutes les applications des RFID, il en est une qui a longtemps utilisée ; c'est l'attribution d'un code individuel à chaque objet. Les principaux groupes industriels viennent d'adopter une norme en ce sens, l'Electronic Product Code (EPC). 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. En fait, l'EPC permet de numéroter tous les objets. Cette norme a été élaborée par l'Auto-ID Center (un centre de recherche) et des laboratoires dans les plus prestigieuses universités du monde. Les laboratoires continuent désormais leurs recherches sous un autre nom, tandis que le Auto-ID Center est devenu EPCglobal, une institution dont le but est de promouvoir et d'encourager l'utilisation de la norme EPC [1] [18].

Comme le code à barres EAN, l'EPC (figure I.5) est un nombre. Il est plus long que le code à barres EAN, au lieu d'être imprimé sous la forme de barres parallèles, il est stocké dans un tag RFID, est un numéro unique attribuable a chaque objet, ce code peut servir à l'obtention d'informations (statuts, localisation,…) via le réseau EPC network [1].

- **La structure du code EPC 96bits :**

01 .	0000A89 .	00016F .	000169DC0.
Header	**EPC Manager**	**Object Class**	**Serial Number**
8 bits	**28 bits**	**24 bits**	**36 bits**

Figure I.5 : Code EPC version 96 bits. [1]

Header : version du standard EPC utilisé
EPC Manager : 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 Quelques exemples d'applications de la technologie RFID dans la vie quotidienne

De nos jours, la technologie RFID est utilisée dans une grande variété d'applications telles que [19] [20]:

I.8.1 La RFID dans les documents d'identité

L'identification des individus passe aussi par l'authentification des papiers d'identité comme montré dans la figure I.6 par exemple dans les nouveaux passeports biométriques possèdent une puce sans contact qui contient des informations telles qu'une photo d'identité, l'empreinte digitale. La RFID est alors un moyen d'une part de s'assurer de la validité des documents d'identité, et aussi de l'intégrité des informations contenues dans ces documents. Il est à noter que des questions de sécurité importantes se posent quant à l'intégrité des données contenues

dans les tags RFID des passeports. Des hackers ont, dès 2006, rendu public les failles de sécurité qui pèsent sur les passeports équipés de RFID.

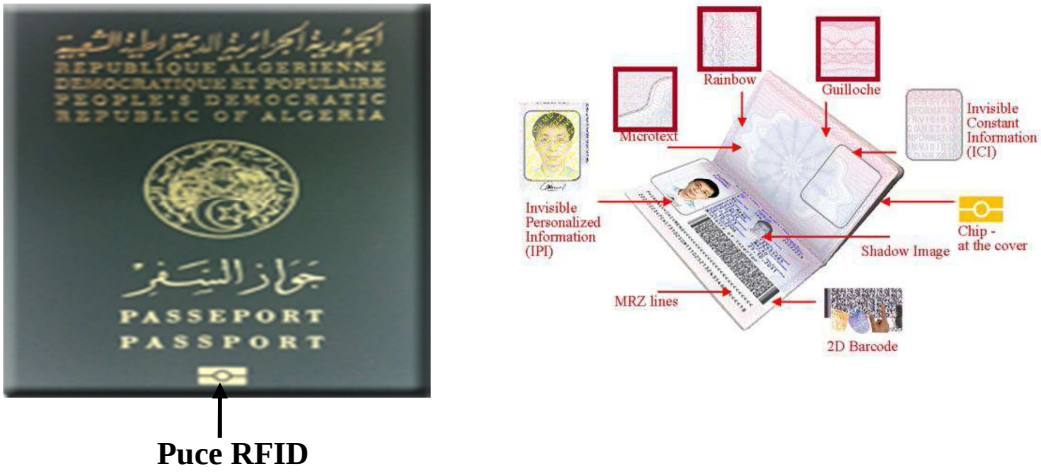


Figure I.6 : La RFID dans les documents d'identité.

1.8.2 La RFID comme outil de paiement :

Parmi les applications de la RFID dans ce domaine, le modèle Pidion BIP-1300 montré par la Figure 1.7, se présente comme un dispositif de type PDA (Personal Digital Assistant), qui accepte des techniques de paiement diverses, comme les cartes bancaires à puce. Il est également équipé d'un module RFID permettant la lecture des cartes sans contact via la connexion WiFi.

Le paiement sans contact peut aussi s'effectuer via une montre. C'est ce que propose Laks, une société autrichienne, depuis 2002. Pour ce faire, la montre accepte une carte SIM personnalisable qui contient la puce RFID. Depuis le début 2009, Laks propose cette montre au public en partenariat avec Austria Card, et avec l'accord de MasterCard.

Figure I.7 : Le dispositif de paiement Pidion BIP-1300.

I.8.3 La RFID dans le secteur de la santé

Dans le domaine de la santé, la localisation des patients et le suivi des processus de soins trouvent une synergie toute particulière grâce aux applications rendues possibles par la RFID.

Les patients sont ainsi équipés de tags qui permettent à la fois de les situer précisément dans l'établissement, et de vérifier que les parcours de soins sont correctement effectués comme le montre la figure I.8. Le contrôle et la surveillance des patients peuvent également s'effectuer par le biais de puces RFID attachées ou implantées dans le corps humain de manière sous cutanée en vue d'assurer leur suivi médical.

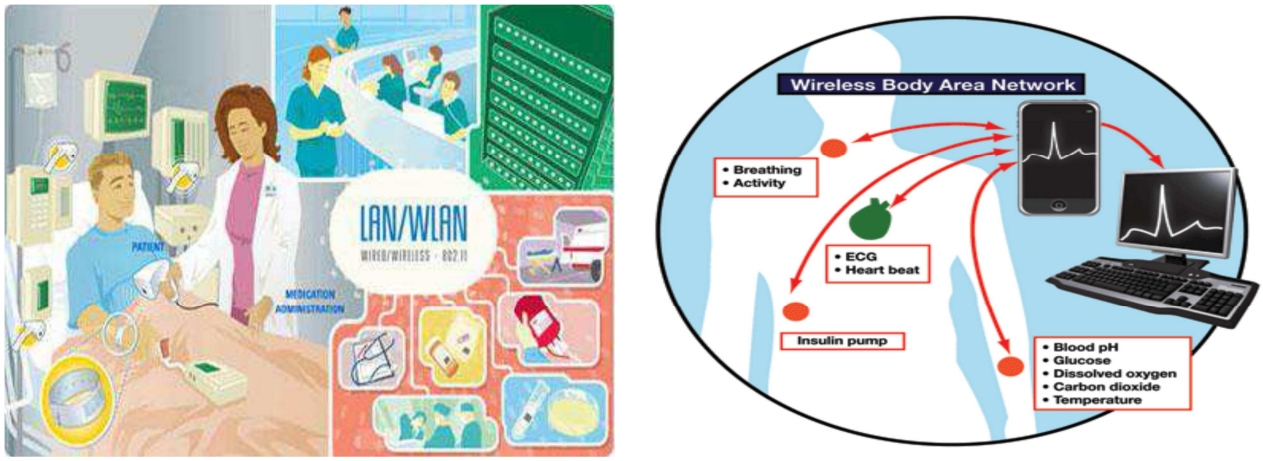


Figure I.8 : La RFID dans le domaine de la santé.

1.9.4 La RFID pour les animaux domestiques

Les animaux d'élevage peuvent être munis d'une étiquette RFID, la société Sniftag propose d'adapter la technologie RFID pour les besoins des animaux de compagnie et de leur maîtres. Accroché au collier d'un animal, un boitier contenant un tag RFID permet de constituer des historiques des différentes activités, de son alimentation, ou encore de son état de santé comme dans la figure I.9.

Figure I.9 : La RFID dans le secteur d'agriculture.

I.8.5 La RFID dans la logistique

Un des secteurs d'activité dans lequel la technologie RFID est utilisée depuis longtemps est celui de la logistique comme montré dans la figure I.10. Rapidement, les codes à barres ont été remplacés par les tags RFID, et dans des domaines aussi divers que la fabrication de médicaments, ou encore dans celui de la gestion de marchandises. L'avantage de la RFID est qu'elle a une portée qui permet de repérer des tags actifs à plusieurs centaines de mètres. Cela permet par exemple d'effectuer des inventaires et de surveiller les marchandises en temps réel.

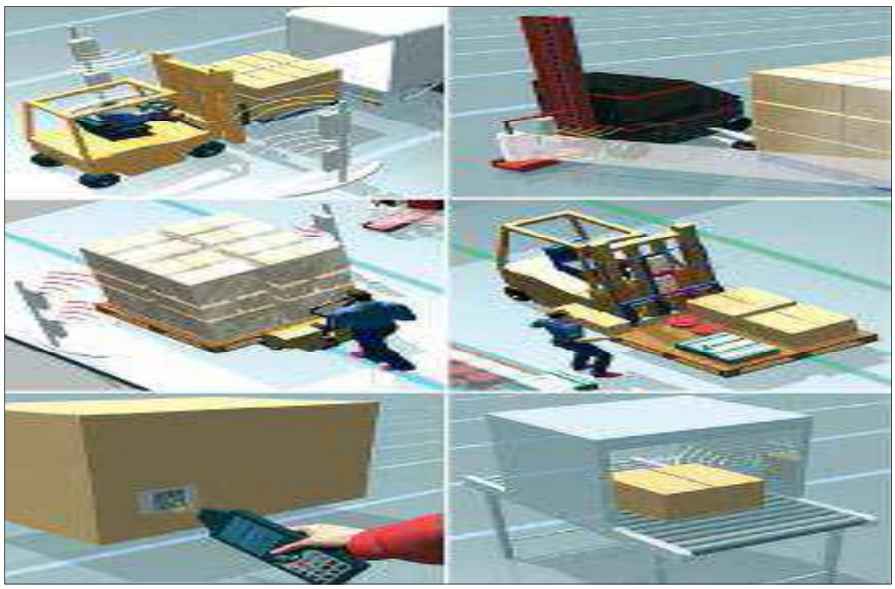


Figure I.10 : La RFID dans le secteur de la logistique.

I.9 Les avantages et inconvénients

La technologie RFID a ses avantages et ses inconvénients, présentons dans ce qui suit les plus essentiels [12] [20]:

I.9.1 Avantages

- Les étiquettes RFID ont une grande capacité de contenu, dans une étiquette radiofréquence une capacité de 1000 caractères est aisément stockable sur $1mm^2$, et peut atteindre 10000 caractères.

- Comme tout support numérique, l'étiquette radio fréquence peut être protégée par mot de passe en écriture ou en lecture. Les données peuvent être chiffrées. Dans une même étiquette, une partie de l'information peut être en accès libre, et l'autre protégée. Cette faculté fait de l'étiquette RFID, un outil adaptée à la lutte contre le vol et la contrefaçon.

- Les étiquettes RFID ont une grande durée de vie, dans les applications où un même objet peut être utilisé plusieurs fois, une étiquette radio fréquence peut être réutilisée 1 000000 de fois.

- Avec l'étiquette radio fréquence, il est possible de s'abstraire des contraintes liées à la lecture optique, elle n'a pas besoin d'être vue. Il lui suffit d'entrer dans le champ de la lecture pour que sa présence soit détectée.

- Les étiquettes RFID n'ont pas besoin d'être positionnées à l'extérieur de l'objet à identifier. Elles peuvent donc être mieux protégées des agressions liées aux stockages, aux manutentions ou au transport. De plus leur principe de fonctionnement ne les rend pas sensibles aux souillures, ou taches diverses qui nuisent à l'utilisation du code à barres.

I.9.2 Inconvénients

- Les prix des étiquettes RFID restent nettement supérieurs à ceux des étiquettes code à barres pour des unités consommateurs. Utiliser les étiquettes radio fréquence à la place du code à barres sur les produits de grande consommation, n'est donc pas aujourd'hui économiquement réaliste. Cela le devient pour lutter contre le vol ou la contrefaçon sur les produits à forte valeur ajoutée. Par contre au-delà du conditionnement unitaire, le coût de l'étiquette radio fréquence peut devenir marginal par rapport à la valeur des produits contenus.

- La perturbation par l'environnement physique, la lecture des étiquettes radio fréquences est perturbée par la présence, par exemple, de métaux dans leur environnement immédiat. Des solutions doivent être étudiées au cas par cas pour minimiser ces perturbations.

- Les perturbations induites par les étiquettes entre elles, dans de nombreuses applications, plusieurs étiquettes radio fréquences peuvent se présenter en même temps dans le champ du lecteur volontairement ou involontairement. Ceci peut être voulu en magasin, au moment du passage à la caisse ou entre les portiques antivol.

- Les systèmes de lecture RFID sont dans certaines circonstances sensibles aux ondes électromagnétiques parasites émises par des équipements informatiques (des écrans d'ordinateurs) ou des systèmes d'éclairages plus généralement par les équipements électriques. Leur emploi doit donc être testé en tenant compte de l'environnement.

- Les interrogations sur l'impact de la radio fréquence sur la santé, cette question fait débat depuis quelques années, en particulier concernant les portiques antivol et les téléphones portables. Les étiquettes passives ne présentent aucun risque quel que soit leur nombre puisqu'elles ne sont actives que lorsqu'elles se trouvent dans le champ d'un lecteur. Les études portent donc essentiellement sur les lecteurs et visent à définir les critères de régulation de leur puissance d'émission afin d'éviter qu'ils ne créent des perturbations sur les équipements de santé tels que les pacemakers (stimulateur cardiaque), mais aussi sur l'organisme humain.

I.10 Quelques défis de la technologie RFID

Comme tout système, la RFID n'est pas sans défis. Nous allons présenter les principaux dans les paragraphes qui vont suivre [21]:

Les systèmes RFID sont particulièrement sensibles aux perturbations liées à l'environnement, il est donc nécessaire de mettre en œuvre des méthodes augmentant la robustesse de ces systèmes. Toutefois il s'agit de systèmes communicants complexes et très hétérogènes dont les performances sont étroitement liées à l'environnement (perturbations, protocole, nombres de tags dans le champ du lecteur…), il est donc difficile d'évaluer les effets des contre-mesures proposées lors de la phase de conception.

Des travaux sur la mise en œuvre de simulateur permettant la modélisation et la simulation de systèmes RFID complexes (middleware, lecteurs et tags) et aussi la réalisation de plateforme de prototypage rapide permettant la validation de nouvelles architectures de tags RFID sécurisés dans un environnement physique réel sont nécessaires. Ces outils sont essentiels pour évaluer des scénarios d'attaque mais aussi les contre-mesures logicielles et matérielles.

Dans les systèmes RFID, il est indispensable de développer des briques matérielles cryptographiques respectant les contraintes de très faible consommation et de très faible surface de ces derniers. Ce travail doit être couplé au développement de la cryptographie légère. Il est donc important de: réduire la surface (partage de ressources entre co-processeurs), éventuellement développer de la cryptographie légère (lightweight cryptographie), de développer des contremesures légères (en s'appuyant éventuellement sur le point précédent, un nouvel algorithme peut certainement être spécifié pour résister mieux ou à moindre coût aux attaques), et enfin définir des protocoles sécurisés qui marchent en environnement contraint.

I.11 Conclusion :

Dans ce chapitre, nous avons présenté les systèmes RFID : les composants, le principe de fonctionnement, quelques normes et standard, Les gammes de fréquence, et quelques domaines d'application, les avantages et inconvénients, et ainsi quelques défis de cette technologie.

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 entamer les problèmes de sécurité dans les systèmes RFID et les contre-mesures mises en place pour battre les attaques sur les systèmes RFIDs.

CHAPITRE II : La sécurité dans les systèmes RFID

II.1 Introduction

La technologie d'identification automatique par radio fréquence est encore à ses balbutiements, mais son utilisation a balayé divers domaines de la vie, et cela à causer plusieurs dommages sur la vie privée. Pour remédier à ce problème, plusieurs solutions de sécurités doivent être employées pour garantir la sécurité de l'information et empêcher les pirates d'accéder aux données sensibles.

Le présent chapitre vise à donner un aperçu des attaques auxquels cette technologie est confrontée et les mesures de sécurité qui ont été mises en œuvre.

II.2 Propriétés de sécurité

La sécurité des systèmes informatiques en général repose sur trois propriétés essentielles: la confidentialité, l'intégrité et la disponibilité. L'ISO a défini six propriétés de sécurité. L'interprétation de ces aspects varie suivant le contexte dans lequel elles sont utilisées. Cette représentation est liée aux besoins des utilisateurs, des services et des lois en usage. La définition et l'application de ces propriétés font partie intégrante des critères d'évaluation de la sécurité [22] [23] [24] [25].

- ***Confidentialité*** **:** pour assurer que seuls les utilisateurs autorisés, dans les conditions normalement prévues, aient accès aux données. Cet accès inclut la lecture, l'impression, la visualisation ou simplement révéler l'existence. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique.

- ***Authenticité*** **:** pour assurer que l'origine d'un objet soit parfaitement identifiée, avec l'assurance que l'identité n'est pas fausse.

- ***Intégrité*** **:** pour assurer qu'une information ne soit modifiée que par les utilisateurs autorisés, dans les conditions normalement prévues. La modification inclut l'écriture, le changement de statut, l'effacement et la création. L'intégrité peut être garantie par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique.

- ***Non-répudiation de l'information(ou l'imputabilité)*****:** pour assurer que ni l'émetteur, ni le récepteur d'un message ne puisse nier la transmission. Ceci peut donc nécessiter l'intégrité, l'accessibilité et la traçabilité des objets, c'est à dire la garantie que les accès et tentatives d'accès aux objets considérés sont tracés et que ces traces sont conservées, exploitables et accessibles, un critère connexe à ceux de traçabilité et d'accessibilité est celui de transparence[1].

- ***Disponibilité*****:** pour assurer qu'un système puisse être utilisé par les utilisateurs autorisés, dans les conditions d'accès et d'usage normales.

[1] Une transmission est transparente si elle ne modifie pas les informations transmises. La transparence peut être sémantique ou temporelle.

- ***Contrôle d'accès :*** Dans le contexte de la sécurité, le contrôle d'accès est la faculté de limiter et de contrôler l'accès à des systèmes et des applications via des maillons de communications. Pour accomplir ce contrôle, chaque entité essayant d'obtenir un accès doit d'abord être authentifiée, ou s'authentifier, de telle sorte que les droits d'accès puissent être adaptés à son cas.

II.3 Attaques de sécurité

Une attaque de sécurité est définie comme toute action compromettant la sécurité de l'information, C'est-à-dire une atteinte à l'une des propriétés de la sécurité. Les objectifs de chaque attaque peuvent être très différents. Il est important d'identifier les cibles potentielles pour comprendre toutes les attaques possibles. La cible peut être le système complet (perturber l'ensemble d'un système d'entreprise) ou seulement une partie de l'ensemble du système (un élément particulier). Les attaques sont classées en deux types génériques sont :

- ***Le premier type* :** dans cette classification, les attaques sont classées selon leurs

a. Attaques passives : Consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible. Deux catégories essentielles d'attaques passives.

b. Attaques actives : Entraine la modification de l'information ou la création de fausses informations. Il est difficile d'empêcher les attaques actives de façon absolue à moins de protéger physiquement tous les moyens et chemins de communications en même temps. Quatre catégories essentielles d'attaques actives.

- ***Le deuxième type* :**

a. Attaques internes : Ce type d'attaques est causé; soit par les utilisateurs autorisés du système qui essayent d'utiliser des privilèges complémentaires dont ils n'ont pas le droit. Soit par les utilisateurs autorisés qui emploient incorrectement les privilèges dont ils ont le droit.

b. Attaques externes : Ce type d'attaques est causé par des utilisateurs externes qui essayent d'accéder à des informations ou des ressources d'une manière illégitime et non autorisée.

II.4 Mécanismes de sécurité

Se sont les mécanismes qui sont conçus pour détecter, prévenir et lutter contre une attaque de sécurité. Elles représentent aussi les objectifs de sécurité. Ils existent plusieurs mécanismes de sécurité parmi eux:

- ***Cryptographie (chiffrement):*** La cryptographique signifie écriture cachée, qui est l'ensemble des techniques (algorithmes, matériels, logiciels) qui repose sur des fondements mathématiques très solides et permettent de réaliser des crypto-systèmes difficiles à casser dans des délais raisonnables [26]. La Cryptographie joue un rôle important pour assurer la

sécurité et la protection de la vie privée dans les systèmes, il existe deux types de cryptographie :

a. **Cryptographie symétrique:** également dite à clé secrète ou à clé privée est la plus ancienne forme de chiffrement et consiste à utiliser la même clé pour le chiffrement et le déchiffrement**.**

b. **Cryptographie asymétrique:** dite aussi cryptographie à clé publique, c'est une approche absolument différente apparue en 1976. Dans cette cryptographie, la clé qui sert à chiffrer est différente de celle qui peut déchiffrer.

- ***Signature numérique* :** un des avantages majeurs de la cryptographie à clé publique est qu'elle fournit une méthode permettant d'utiliser des signatures numériques, elles permettent de garantir l'authentification et le contrôle d'intégrité des données et procure également la non répudiation de l'information [27].

- ***Les fonctions de hachage* :** aussi appelée fonction de condensation, une fonction de hachage est une fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe, la chaîne résultante est appelée empreinte (digest en anglais) ou condensé de la chaîne initial. La fonction de hachage est une fonction à sens unique, c'est-à-dire qu'elle doit permettre de trouver facilement l'empreinte à partir du message, et d'empêcher de retrouver le message à partir de l'empreinte. Elle doit aussi être très sensible, pour qu'une petite modification du message entraîne une grande modification de l'empreinte. Autre caractéristique d'une fonction de hachage, est qu'elle doit prendre en charge les collisions et savoir les gérer, c'est-à-dire qu'elle doit empêcher que deux messages différents aient la même empreinte. De tel façon, les fonctions de hachage garanties l'intégrité du message [27] [28].

II.5 RFID : Attaques et Sécurités

II.5.1 Classification des applications RFID selon les besoins de sécurité

En plus de la classification des tags, il est nécessaire de classer les applications en fonction de leurs objectifs. La technologie sans contact, voir les Figures II.1 et II.2, sont majoritairement utilisées pour l'identification, l'authentification et la localisation des personnes et objets (passeport, carte de transport, ...).

Le premier cas (identification), est une phase qui consiste à établir l'identité de l'utilisateur. Elle permet de répondre à la question : "Qui êtes vous ?". L'utilisateur utilise un identifiant (que l'on nomme "Compte d'accès", "Nom d'utilisateur" ou "Login" en anglais) qui l'identifie et qui lui est attribué individuellement. Cet identifiant est unique [29].

Identification = Obtenir l'identité du tag.

Le deuxième cas (authentification), est une phase qui permet à l'utilisateur d'apporter la preuve de son identité. Elle intervient après la phase d'identification. Elle permet de répondre

à la question : "Êtes-vous réellement cette personne ?". L'utilisateur utilise un authentifiant ou "code secret" que lui seul connait [29].

Authentification = Obtenir l'identité du tag + preuve du tag

Le troisième cas (localisation), consiste à situer une étiquette RFID. Pour cela, il est possible d'utiliser un réseau de lecteurs pouvant communiquer entre eux ou avec un système central pour localiser les étiquettes RFID. En effet, la corrélation des signaux d'une étiquette, reçus par différents lecteurs, permet de la situer précisément. Une autre possibilité est tout simplement de situer plus approximativement l'étiquette comme étant dans le champ d'action d'un lecteur donné. Cette dernière solution est utilisée par exemple dans certains aéroports pour localiser les bagages enregistrés [7].

Figure II.1 : Authentification par RFID.

Figure II.2 : Identification par RFID.

II.5.2 Types d'attaques et contre-mesures dans les systèmes RFID

Comme tous d'autres systèmes d'information, les systèmes RFID sont vulnérables aux attaques et peuvent être compromis à différents stades de leurs utilisations [11]. Les attaques contre un système RFID peuvent être passives ou actives et sont classées généralement en quatre grands groupes: les attaques sur l'authenticité, atteintes à l'intégrité, les attaques contre la confidentialité et les attaques sur la disponibilité. En plus d'être vulnérables à l'ensemble des attaques telles que l'écoute de man-in-the-middle, et déni de service. La technologie RFID est en particulier sensible à l'usurpation d'identité. Cette section illustre les différents types d'attaques et fournit des contre - mesures pour ces dernières.

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. [30] 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. [31] 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. [32] 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.

Les auteurs de [33] ont proposé une classification de 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. [34] 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. [35] ont divisé les menaces RFID en quatre couches principales: la couche physique, La couche transport, la couche application et de la couche stratégique. En outre, ils ont créé une catégorie distincte d'attaques multicouches qui exploitent les vulnérabilités à partir de plusieurs couches.

Mitrokosta et al. [36] ont spécifié trois catégories principales des attaques RFID basées sur la partie du système qu'ils visent (les attaques affectes à la couche de matériel RFID, la couche de communication, 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).

Les auteurs dans [37], 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.

Dans cette partie on va utiliser la classification proposée par les auteurs de [33] :

II.5.2.1 Menaces de la couche physique : la couche physique dans les communications RFID est constituée de l'interface physique, des signaux radios utilisés, et des matériels RFID. Les attaquants de cette couche profitent de la nature sans fil des communications RFID, de leurs faibles protections physiques et leurs manques en termes de résistance contre les manipulations physiques.

II.5.2.1.1 Skimming ou Activation à distance

Le principe de cette attaque est d'activer et lire une carte sans l'autorisation de son propriétaire, voir Figure II.3. Cette attaque est active. En général, cette attaque permet de communiquer avec la carte en dehors de sa plage de fonctionnement. La distance de fonctionnement nominal d'un système conforme à la norme ISO14443 est d'une dizaine de centimètres, l'attaquant doit donc être capable de communiquer avec la carte à une distance supérieure. Les deux principales difficultés liées à cette attaque sont la télé-alimentation de la carte et la récupération des données envoyées par cette dernière. La distance d'activation de la carte est donc énormément limitée par ces paramètres [16].

L'objectif majeur de cette attaque est la récupération des données contenues sur le tag, mais ses applications sont multiples (attaque replay, clonage, déni de service,…). Cette attaque reste difficile à réaliser, l'attaquant doit être capable d'alimenter la carte tout en modulant son champ pour envoyer les commandes. Un compromis est important entre distance d'activation et récupération des données.

Les attaques skimming sont à craindre car ils peuvent faire partie d'une bombe RFID. Autrement dit, une bombe qui analyse son rayon d'explosion pour les étiquettes RFID et seulement explose lorsque les données stockées sur l'ensemble des étiquettes dans son voisinage remplissent une liste de demandes. Il pourrait par exemple être prêt à exploser quand une certaine personne passe ou lorsqu'au moins 10 personnes d'une nationalité donnée sont dans son rayon d'explosion [38].

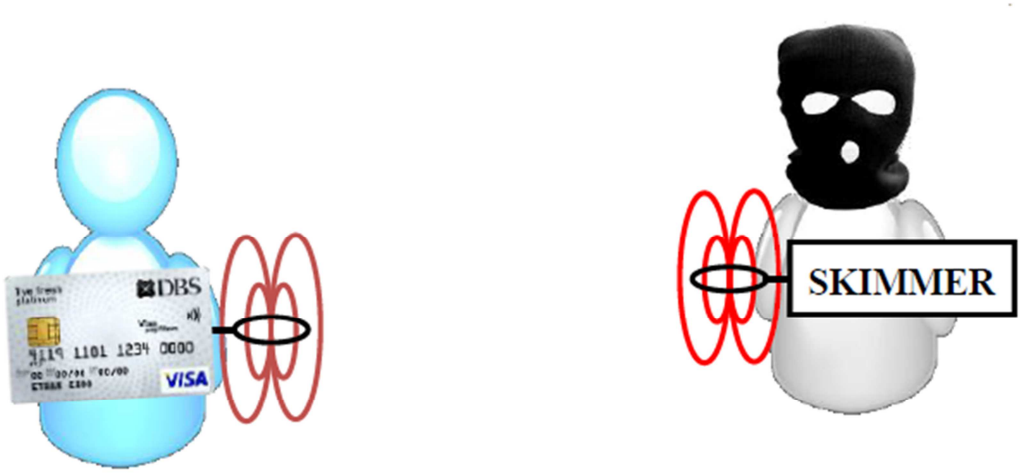


Figure II.3 : Attaque Skimmig [16]

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 évitéé.
Des exemples de mécanismes d'authentification pour la RFID sont : authentification symmetric et Hash-lock (verrouillage du tag).

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 tels que: Blocker tag[2] et RFID Guardian[3] :

II.5.2.1.2 Ecoute à distance (Eavesdropping ou surveillance non autorisée)

Une attaque eavesdropping, voir Figure II.4, consiste à écouter une transaction privée à distance entre un lecteur et un tag sans l'accord des participants dans l'intention d'y révéler des secrets. Cette attaque est passive. Cette attaque au niveau de la couche physique est l'une des plus simples à réaliser, car elle ne requiert que très peu de matériel. Une simple sonde de champ[4] et un oscilloscope[5] peuvent suffire à récupérer des signaux à plusieurs mètres de distance [16]. Les trames enregistrées lors d'une attaque eavesdropping vont servir à récupérer des informations plus ou moins importantes. Elles peuvent aussi servir dans le cadre d'une attaque clonage, spoofing ou replay (voir II.5.2.1.3) qui sont souvent précédées d'une attaque eavesdropping permettant à l'attaquant de récupérer les données enregistrées dans la mémoire d'un transpondeur (tag).

Figure II.4 : Attaque Eavesdropping [16]

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, les espionnant peut être en mesure d'entendre la communication, mais pas la déchiffrer.
Ce cryptage permet aussi d'inclure la protection contre le skimming en exigeant la connaissance d'une clé secrète pour déchiffrer les messages provenant d'une étiquette.

[2] Blocker tag: est un dispositif électronique utilisé par des personnes possédant des tags.
[3] RFID Guardian: est un transpondeur intelligent permettant de bloquer les tags sélectionnés en utilisant les algorithmes d'anticollision de l'ISO14443-A.
[4] Sonde de champ: est une appareille utilisée pour l'exploration du champ.
[5] Oscilloscope: appareil qui permet de visualiser les variations instantanées d'une grandeur physique grâce à une courbe.

II.5.2.1.3 Clonage (cloning), spoofing (usurpation), replay

Ces attaques sont regroupées dans la même partie, car leurs principales caractéristiques sont les mêmes. Toutes ces attaques nécessitent la récupération malhonnête des données sur une autre carte sans contact. Ces attaques sont souvent précédées d'une attaque eavesdropping ou d'une attaque skimming permettant à l'attaquant de récupérer les données enregistrées dans la mémoire d'un tag. Ces données peuvent alors être enregistrées sur un tag vierge pour obtenir une copie de la carte précédemment attaquée.

Ces types d'attaques arrivent à vaincre le contrôle d'accès [39], permettant ainsi l'accès non autorisé à la vie privée et aux systèmes tel que l'e-paiement.

Contre-mesures :

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

a. 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, peuvent être introduites dans le cadre d'une information correcte. 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". Cependant, de nombreux systèmes reposent sur le secret des algorithmes et des protocoles pour améliorer la sécurité fournie par la cryptographie, et donc se contenter de longueurs de clés courtes.

b. Pour vaincre les attaques physiques de clonage, les auteurs de [40] ont proposé d'utiliser les fonctions physiques inclonable[6] (PUFs) de mémoire sécurisée pour le stockage de la clé secrète sur un tag RFID. Les auteurs ont affirmé que l'attaque de clonage peut être évitée.

c. La solution la plus utilisée pour prévenir les attaques par rejeu est l'utilisation d'un mécanisme de contestation (opposition) et de réponse. les systèmes à base de temps et à base de compteur peuvent également être utilisés comme des contre-mesures pour les attaques par rejeu.

II.5.2.1.4 Imitation du lecteur RFID

Dans les systèmes RFID, beaucoup de communications RFID n'ont pas besoin d'authentification. Dons les attaquants peuvent facilement imiter l'identité d'un lecteur légitime afin d'obtenir ou modifier des données présentes sur les étiquettes RFID connectées au lecteur. La possibilité de ces attaques repose sur les mesures de sécurité employées pour

[6] Les fonctions physiques inclonable (PUFs) sont aujourd'hui largement reconnues comme étant des primitives de sécurité essentielles, parce qu'elles permettent d'équiper les systèmes embarqués d'une biométrie du silicium. Grâce à cette technologie, les secrets et données personnelles n'ont plus à être stockés dans une mémoire non volatile mais peuvent être reconstruits quand besoin est à partir d'une propriété électronique unique de la puce.

l'authentification du lecteur RFID. Il est alors plus ou moins facile, voire impossible, d'effectuer ces attaques.

Contre-mesures:

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

II.5.2.1.5 Attaque side-channels

Les attaques side-channels (fuite d'informations), sont des attaques reposant sur la fuite d'informations du tag. Elles consistent à espionner l'activité interne de la puce RFID afin de découvrir des informations secrètes (clé secrète, algorithme de cryptage). Seule l'attaque RFA[7] est spécifique au lien sans contact (RFID).

Contre-mesure :

Pour éviter la collecte secrète d'informations, il est envisageable d'agir directement au niveau transport, en bloquant ou en régulant finement les communications de la puce RFID.

Par exemple selon le principe de la cage de Faraday, il est possible d'empêcher le fonctionnement d'un système RFID en enveloppant l'étiquette RFID dans un conteneur recouvrit de métal. L'utilisateur de puces RFID peut en utilisant cette technique se protéger des requêtes indiscrètes sur celle-ci.

II.5.2.1.6 Déni de service (DoS)

Les attaques DoS peuvent prendre différentes formes en attaquant l'étiquette RFID, le réseau ou la base de données backend, le but est de ne pas voler ou de modifier des informations, mais pour désactiver le système RFID de sorte qu'il ne peut pas être utilisé. Plusieurs attaques de déni de service existant dans les systèmes RFID, dans cette partie on va présenter deux attaques :

a. Brouillage (Jamming): Dans ce type d'attaque, l'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 [41].

L'attaquant dans le brouillage doit émettre un champ magnétique dans la même bande de fréquence que le lecteur avec une bande passante au moins égale et un champ magnétique très important. La puissance du champ émis par un lecteur est limitée par une norme européenne: la législation ETSI[8] (Figure II.5). La valeur du champ magnétique à 10 m ne doit pas dépasser

[7] RFA (pour RadioFrequency Analysis): Enregistrement du rayonnement électromagnétique émis en dehors des échanges par une carte sans contact au travers du canal normal c'est-à-dire par son antenne.

[8] ETSI (pour European Telecommunications Standards Institute), c'est-à-dire l'institut européen des normes de télécommunications, est l'organisme de normalisation européen du domaine des télécommunications.

42dBµA/m. Si l'attaquant développe un dispositif permettant d'obtenir un champ magnétique d'amplitude supérieure à celle imposée par la législation, il est sûr de brouiller le signal du lecteur [16].

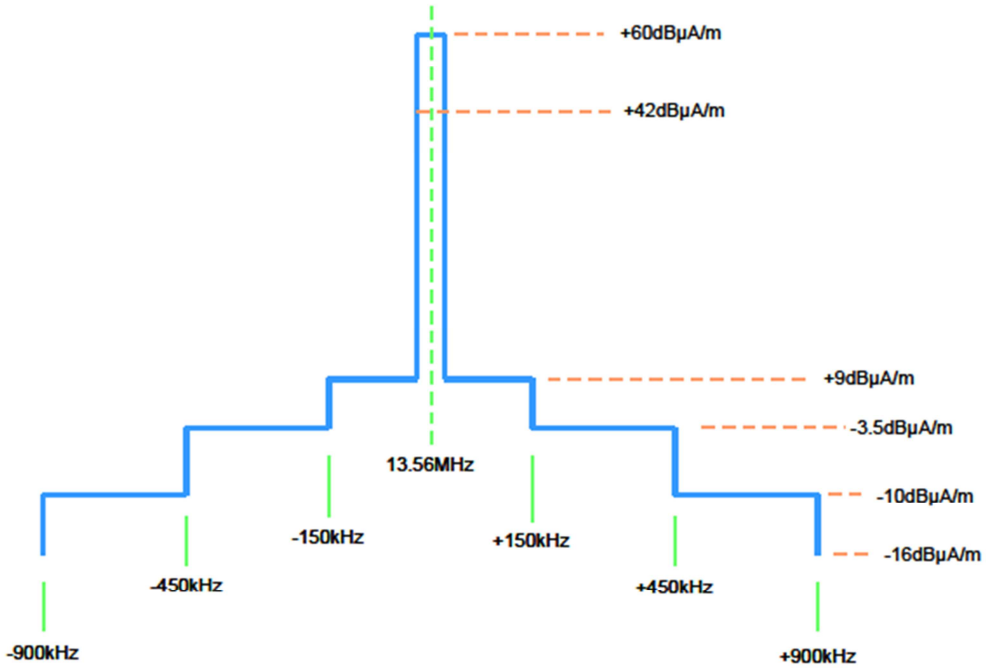


Figure II.5 : champ magnétique maximum à 10 m du lecteur [16]

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 [16]. Les signaux transmissent peuvent être bloquées ou diminuer en utilisant un système de cage de faraday, voir Figure II.6. 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 [41].

Figure II.6: Cage de faraday bloquent les signaux dans la gamme de fréquences 10MHz-20GHz.

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 détecté, les attaques peuvent généralement être arrêtées avant qu'ils font trop de mal.

Une des contre-mesures utilisée pour les attaques Dos est les protocoles d'anticollision qui sont des algorithmes permettant d'éviter que plusieurs tags communiquent en même temps.

II.5.2.1.7 Attaque par relais (Relay)

Les applications sensibles, comme le contrôle d'accès, reposent généralement sur le standard ISO 14443 qui offre une distance de communication relativement faible, de l'ordre de quelques centimètres. La distance de communication est en effet la pierre angulaire de la sécurité en RFID passive. Le tag répondant à toute invitation d'un lecteur sans demander l'accord de son porteur, l'accord implicite de celui-ci est sa présence dans le champ du lecteur. Cela implique une distance de communication courte [16].

Reposant sur ce fait, l'objectif de l'attaque relais consiste à établir une communication entre un lecteur et une carte sans contact sans l'accord de son propriétaire. La distance entre le tag et le lecteur est souvent supérieure à la distance nominale de fonctionnement, le lecteur est pourtant convaincu que la carte sans contact est dans son champ RF.

En 2005, Gerhard Hancke est passé avec succès de la théorie à la réalisation pratique d'un projet démontrant une attaque par relais sur des tags RFID de la norme ISO 14443. Sa démonstration était applicable sur une distance pouvant aller jusqu'à 50 mètres entre le tag et le lecteur, car il repose sur une technique de relais du canal physique, avec une execution d'une simple modulation/démodulation par des équipements pirates placés à proximité des deux extrémités [42]. Les dispositifs permettant de réaliser cette attaque, voir Figure II.7, est constitué de deux éléments : un môle (lecteur simulé) et un proxy (tag simulé). Ceux-ci peuvent être reliés entre eux par un lien filaire ou sans fil. Ensemble, ces deux éléments vont relayer les informations entre le lecteur et la carte.

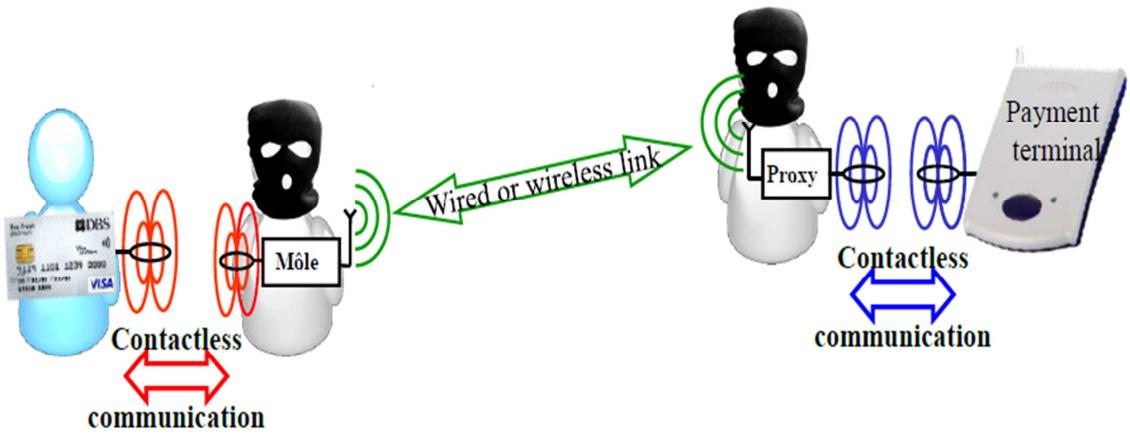


Figure II.7 : Présentation d'une attaque relais sans fil proposée par G.Hancke [42].

Le dispositif « môle» doit être placé assez près de la carte de la victime. Il peut être assimilé à un système de skimming, il a trois fonctions principales :

- Se mettre à la place du véritable lecteur et envoyer les signaux transmis par le proxy à la carte.
- Télé-alimenter de la carte sans contact.
- Transmettre les réponses de la carte au proxy.

Le dispositif « proxy» est placé près de lecteur, il doit se comporter comme une carte sans contact valide, il a deux fonctions :

- Transmettre les requêtes du lecteur au môle.
- Récupérer les données envoyées par le môle et les transmettre au lecteur de manière transparente.

Cette attaque est possible car une étiquette RFID peut être interrogée sans que son possesseur ne s'en rende compte. On peut alors très vite imaginer un scénario d'attaque, ou une personne mal intentionnée tente d'entrer dans un bâtiment, aidé d'un associé se trouvant à proximité d'une étiquette ayant les autorisations nécessaires (un membre du personnel de l'établissement se trouvant dans les transports en commun, ayant son étiquette RFID dans sa poche par exemple). Il suffit alors que la personne mal intentionnée et son associé s'arrangent pour faire transmettre le défi, et le lecteur se fera tromper [43].

Contre-mesures :

De se protéger contre une attaque par relais n'est pas facile, comme l'a déclaré Gerhard Hancke [42]; une attaque de relais n'sont pas vraiment une violation des protocoles cryptographiques constituant un mécanisme d'authentification.

Un protocole de Distance Bounding introduit dans [44] est un algorithme de cryptographie qui permet au lecteur de s'assurer que le tag avec lequel il communique se trouve dans sa zone de couverture. Ce type de protocole est une des seules contre-mesures identifiées pour détecter les attaques relais.

II.5.2.2 Menaces de la couche transport : Les attaques destinées à cette couche met en valeur la manière dont les données sont transmises entre les lecteurs et les étiquettes.

II.5.2.2.1 Attaque de Main In-The-Middle (MITM)

L'une des attaques les plus puissantes contre la technologie RFID est Main-In-The-Middle. C'est une variante de l'attaque relais. En effet, cette attaque possède des caractéristiques similaires à l'attaque relais: l'objectif consiste à transmettre les données entre le lecteur et le tag en passant par le relais (espion), voir Figure II.8. La principale différence de cette attaque est qu'elle peut être active : les informations échangées entre le lecteur et la carte vont être modifiées durant le passage dans le relais, ou passive: écoutant seulement les informations qui transitent [45].

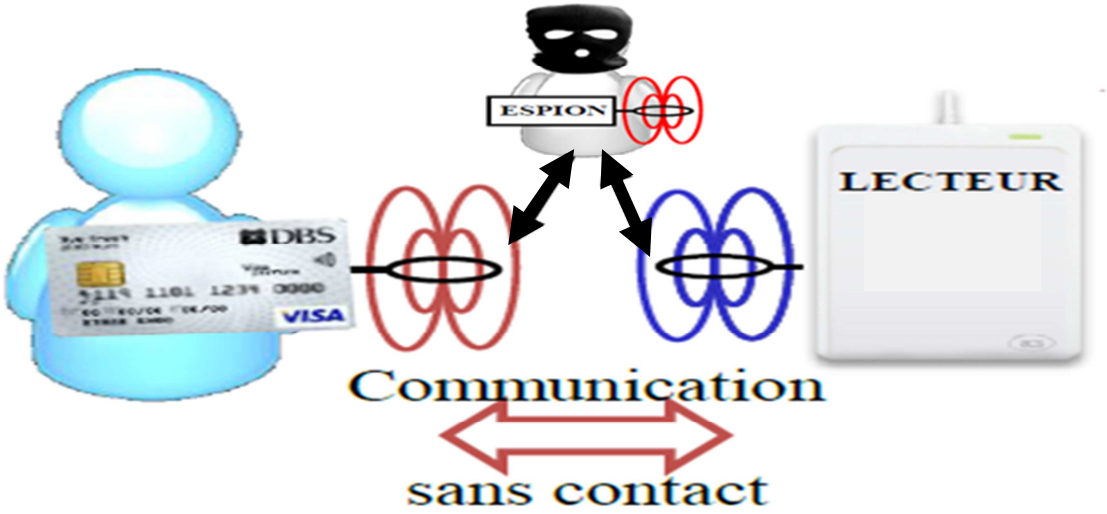


Figure II.8: Attaque main in-the-middle.

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.

II.5.2.2.2 Analyse de traffic

Etant donné que la communication dans les systèmes RFID est sans fil, ce type d'attaque est particulièrement dangereux. Rien n'empêche un espion d'intercepter le trafic entre le tag et le lecteur RFID pour en extraire d'informations. Même dans le cas où la communication est chiffrée, l'écoute du trafic reste utile. En effet, bien que l'on soit incapable de restituer le message dans son intégralité, il est possible de déduire au moins une partie de l'information en se référant à des patterns de communication. En outre, même la détection de la présence ou de l'absence de communication peut être prouvée. Ainsi, en temps de guerre, un silence radio peut très bien signifier une attaque rapprochée. Le même type d'induction peut être appliqué aux communications RFID.

Contre-mesure :

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

II.5.2.3 Menaces de la couche application: Les attaques contre cette couche visent les informations contenues dans les applications et les liaisons entre les utilisateurs et les étiquettes RFID.

II.5.2.3.1 Altération des données

L'altération des données est de bloquer l'émission radio d'une étiquette RFID. A titre d'exemple, 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 mal intentionnée peut obtenir une réduction impressionnante sans évoquer le moindre soupçon. Seul un catalogue physique pourra alors détecter 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.5.2.3.1 Menaces sur les intergiciels (middleware) RFID

Puisque les middlewares RFID incluent des matériels réseaux, un attaquant peut profiter des ressources limitées du système RFID et causer un déni de service au niveau du middleware RFID, en envoyant un nombre important de paquets au middleware pour que le réseau soit inondé. La surcharge des requêtes empêche alors le traitement du trafic normal.

Contre-mesures :

Les attaques de ce type sont généralement très difficiles à empêcher. Il n'existe pas de bons mécanismes pour contrecarrer ces attaques. Cependant, elles sont souvent faciles à détecter, et elles peuvent donc être arrêtées avant qu'elles fassent trop de mal.

II.5.2.4 Autres attaques : Les attaques de ce niveau sont les attaques spécifiques au système.

II.5.2.4.1 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 [46]. 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 127 caractères. 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.

Rieback et al. [46] ont également expliqué comment le virus RFID fonctionne dans une chaîne d'approvisionnement. Si un conteneur est arrivé à un centre de distribution et l'étiquette RFID du conteneur avait été infecté par un virus informatique, ce virus RFID particulier pourrait utiliser l'injection SQL (Structured Query Language) pour attaquer les serveurs principaux et éventuellement mettre le système RFID vers le bas.

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 ya 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.5.2.4.2 Attaques cryptographiques

Les attaques cryptographiques dans les systèmes RFIDs sont largement utilisées, les attaquants peuvent employer ces attaques afin de casser les algorithmes de cryptographie utilisés dans les communications entre les tags et les lecteurs afin d'extraire des informations secrètes.

Contre-mesures

Les mécanismes de défense contre les attaques de cryptographie sont vastes, dans cette partie on cite quelques mécanismes :

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

II.5.2.4.3 Tracking (traçabilité)

La notion fondamentale de la RFID est que chaque tag est équipé d'un ID (identificateur unique). Ainsi, si un attaquant peut lire cet ID, les étiquettes peuvent être reconnues quand elles se déplacent dans le temps et l'espace. Cela peut permettre à des attaquants de suivre les étiquettes. La traçabilité représente une menace contre la confidentialité des données privées [47].

Cependant, la traçabilité est un problème multicouche. Se cacher l'identité de l'étiquette utilisée par la couche d'application ne permet pas d'interdire la traçabilité de l'étiquette sur la base d'une couche inférieure.

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 utilisée.

II.5.3 Résumé des attaques et contre-mesures

En général, il existe de nombreuses façons d'attaquer les systèmes RFID. Il ya aussi de nombreuses façons de se défendre contre ces attaques. Les mécanismes de défense peuvent être soit mis en œuvre dans Le système (les protocoles de sécurité mises en œuvre) ou leurs utilisation peut être laissée aux utilisateurs. Le Tableau II.1 résume les attaques et catégorise les contre-mesures en fonction de celui qui est responsable de leurs mises en œuvre :

Attaques	Influence sur les services de sécurité			Contre-mesures	
	Intégrité	**Confidentia -lité**	**Disponibilité**	**Système (Protocoles)**	**Utilisateurs**
Skimming	oui	oui	oui	-Authentification -Authentification Symétrique -Hash-lock	-Blindage -Blocker Tag -RFID Gardian
Eavesdropping	oui	oui	oui	-Cryptage	
Replay	oui	oui	non	-informations secrètes -Authentification	
Imitation du lecteur RFID	non	oui	non	Chiffrement	
Side-channels	oui	oui	non		-cage de Faraday
Déni de service	non	non	oui	-Contre mesures manuelles -algorithmes déterministes	
Relay	oui	oui	non	-Distance Bounding -Activation séparée par l'utilisateur -Deux facteurs d'authentificati on	-Blindage -RFID Gardian

MITM	oui	oui	non	-Cryptographie -Authentification	
Analyse de traffic	oui	oui	non	Cryptographie	
Altération des données	oui	oui	non	Contrôle d'accès	
Menaces sur les intergiciels	non	non	oui	Pas de bons mécanismes pour l'empêcher	
Virus RFID	oui	oui	oui	-Permission limitée -Bounds checking -Paramètres de liason	
Attaques cryptographiques	non	oui	non	-Authentifica-tion défi / répense -etc...	
Tracking	non	oui	non	-Identifiant aléatoire -Randomisation -commande Kill	-Blindage -RFID Gardian
Cloning	oui	oui	non	-informations secrètes -Authentification	
Spoofing	oui	oui	non	-informations secrètes -Authentification	

Tableau II.1: Résumé des attaques sur les RFID

II.5.4 Respect de la vie privée

Outre les problèmes directement liés à la sécurité, la RFID doit faire face aux problèmes qui touchent la vie privée. Ces problèmes concernent la divulgation d'informations et la traçabilité malveillante… etc.

- *Divulgation d'information*

Le problème de la divulgation d'informations se pose lorsque les données envoyées par le tag révèlent des informations sur l'objet qui le porte, plusieurs types de divulgation peuvent être cités:

- Spam : Il est possible de se faire apposer un « traceur » par une personne mal intentionnée, cette personne peut alors suivre nos déplacements ou obtenir des informations privées sur notre mode de vie, notre personnalité,... Ces données récupérées peuvent être ajoutées dans une base de données de façon à être vendues à des marchands ou autres. Ces informations peuvent ensuite être utilisées pour mieux connaître nos habitudes de consommation et nous proposer des objets en fonction de nos préférences.

- Terrorisme, profilage : Les informations récupérées sur un « tracer » ou sur un transpondeur par espionnage ou activation peuvent être utilisées pour classer ou agresser des personnes selon des critères politiques, ethniques, ...

- *Traçabilité malveillante des individus*

Les tags électroniques n'ont pas appel à contenir ou à transmettre d'importantes quantités de données. Lorsqu'une base de données est présente dans le système, le tag peut envoyer qu'un simple identifiant, que seules les personnes ayant accès à la base de données peuvent relier à l'objet correspondant. C'est le principe utilisé par les systèmes à codes-barres. Cependant, même si un identifiant ne permet pas d'obtenir d'informations sur l'objet lui-même, il peut permettre de le tracer. On peut ainsi savoir à quelle heure une personne est passée en un lieu donné, par exemple pour déterminer son heure d'arrivée et de départ de son poste de travail, ou on peut reconstituer son chemin à partir de plusieurs lecteurs, par exemple dans une entreprise ou un centre commercial.

- *Puce sous-cutanée*

A l'heure de la miniaturisation des systèmes électroniques, l'injection des puces RFID dans le corps est déjà une réalité. A l'état microscopique, il serait possible de tracer toute une population de façon totalement illicite ce qui met en danger la vie privée des individus.

II.5.5 S'attendre à l'inattendu

La technologie RFID est encore jeune, mais des techniques d'attaques nouvelles et inattendues pourraient apparaître. À titre d'exemple, en février 2006, le professeur et célèbre cryptographe Adi Shamir a utilisé une antenne directionnelle et un oscilloscope numérique pour examiner l'énergie utilisée par les puces RFID pendant leur lecture. Les types d'utilisation de l'énergie ont pu être analysés pour déterminer à quel moment la puce recevait des bits de mots de passe corrects et incorrects. M. Shamir a indiqué qu'un téléphone portable possède tous les moyens nécessaires pour mener une attaque et compromettre toutes les puces RFID situées à proximité. Cet exemple montre toutefois qu'il serait souhaitable de tenir compte de l'évolution de la technologie lorsque l'on définit une stratégie en matière de

sécurité, d'évaluer les risques au moment de la conception du système, et de les réévaluer régulièrement comme dans n'importe quel processus de gestion de la sécurité [5].

II.6 Conclusion

En raison du déploiement le plus large des systèmes RFID, les problèmes de sécurité dans ces systèmes devient de plus en plus incontournable. Pour remédier aux problèmes de sécurité relatifs aux systèmes RFID, plusieurs solutions ont été déjà proposées tel que la cryptographie.

Dans ce chapitre, nous avons essayé de découvrir certaines attaques possibles qui peuvent affecter ces systèmes. En considérant le point d'attaque, sa classification et nous avons discuté des contre-mesures et des mesures de sécurité possibles qui peuvent être utilisés pour lutter contre ces attaques.

Le chapitre suivant prend la charge de présenter la cryptographie dans les systèmes RFIDs et de définir quelques protocoles cryptographiques mises en place.

CHAPITRE III : Les protocoles cryptographiques des les systèmes RFID.

III.1 Introduction

Les communications dans les systèmes RFID sont exécutées sur des canaux radio fréquences publics, et tout lecteur peut obtenir des informations sur l'étiquette facilement, ces systèmes sans protection adéquate sont vulnérables à plusieurs attaques. Ces problèmes de sécurité et de confidentialité interdit le déploiement le plus rapides de la RFID. Pour résoudre ces problèmes, de nombreuses solutions sont proposé parmi eux les protocoles cryptographiques qui représentent un grand intérêt pour assurer la sécurité dans toutes les couches RFID: couche physique, couche application et couche transport (où se trouvent les protocoles de sécurité).

Ce nouveau chapitre va ce consacré aux protocoles cryptographiques dans les systèmes RFIDs, dans on va présenter quelques protocoles d'authentification.

III.2 Les protocoles cryptographiques dans les systèmes RFID

L'utilisation des étiquettes RFID peut simplifier de nombreuses applications et fournir de nombreux avantages, mais la vie privée des utilisateurs doit être prise en compte. Une menace potentielle pour la vie privée d'un utilisateur est que n'importe lecteur peut obtenir des informations sur les mots-clés dans le système. 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. 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

Les Protocoles légers ont l'avantage de maintenir la demande de calcul et le coût très faible des étiquettes RFID. Pour cette raison, les protocoles légers ont un intérêt pour l'industrie et le milieu universitaire. La cryptographie lightweight est divisée en deux catégories comme elle la Figure III.1:

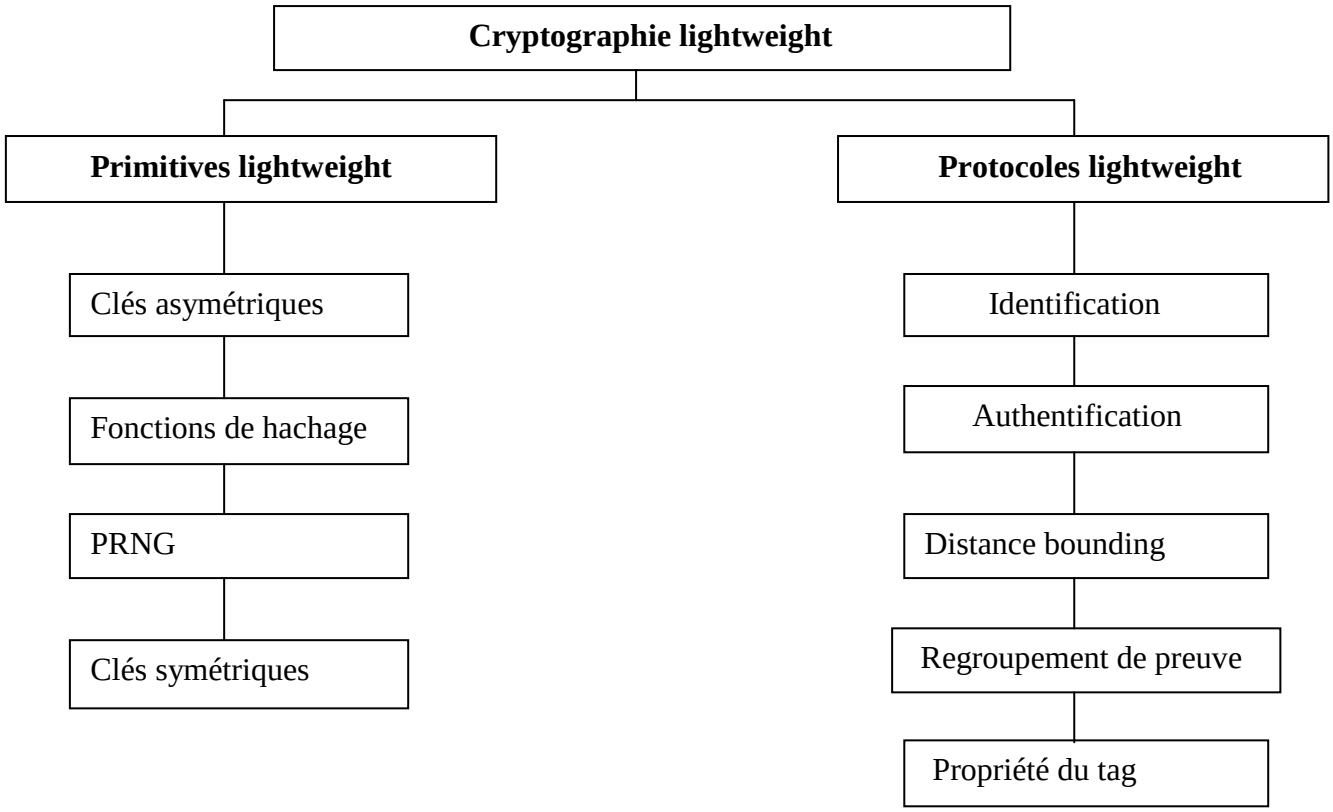


Figure III.1: Les protocoles légers dans les systèmes RFID

➢ Primitives lightweight: cette catégorie regroupe les primitives symétriques, les primitives asymétriques, les fonctions de hachage et les générateurs de nombres aléatoires [48] [49].

➢ 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 [49].

III.2.2 Les protocoles ultralightweight

Parmi les propositions fondées sur des primitives cryptographiques, les protocoles utilisant des fonctions de hachage ou MACs (codes d'authentification du message) sont parmi les premières solutions discutées dans la littérature. Ils existent également des protocoles basés sur PRNGs (Nombres pseudo-aléatoires Générateurs), le chiffrement de flux, les blocs de chiffrement et même clés publiques [49].
D'autre part, des protocoles basés sur les opérations ultralégeres 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 propositions de ce type sont divisées 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

L'importance de la cryptographie dans les systèmes RFIDs est remarquable vu le nombre important de protocoles cryptographiques mise en œuvre dés la naissance de cette technologie. Les protocoles RFID utilisant l'identifiant ID manipulent deux mécanismes sont: statique et dynamique. La particularité du mécanisme d'ID statique est que l'identificateur du tag reste le même pendant l'authentification entière, mais celui du mécanisme dynamique, l'identificateur de tag est modifié. Chaque mécanisme a ses avantages et inconvénients comme tous autres mécanismes.

III.3.1 Le protocole R^2AP (Reconstruction based RFID Authentication Protocol)

La référence [50], présente une mise en œuvre d'un nouveau protocole RFID ultraléger à base de protocole d'authentification nommé R^2AP, qui est basé sur l'utilisation d'une nouvelle opération de reconstruction au niveau du bit. Dans cette section, on vérifie ce dernier protocole. On note que la nomination des protocoles revient aux premiers caractères de nom des auteurs.

III.3.1.1 Notations

Les notations utilisées dans ce protocole sont présentées dans le tableau III.1:

Notations	Signification
R	Lecteur
T	Tag
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
$\oplus$	L'opération xor
Rot(x,y)	Rotation gauche de x par y bits
Rec(x,y)	L'opération de reconstruction de x par y
$\|\|$	Concaténation

Tableau III.1: Notations utilisées dans R^2AP

III.3.1.2 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. Le protocole R2AP possède sept étapes, voir Figure III.2:

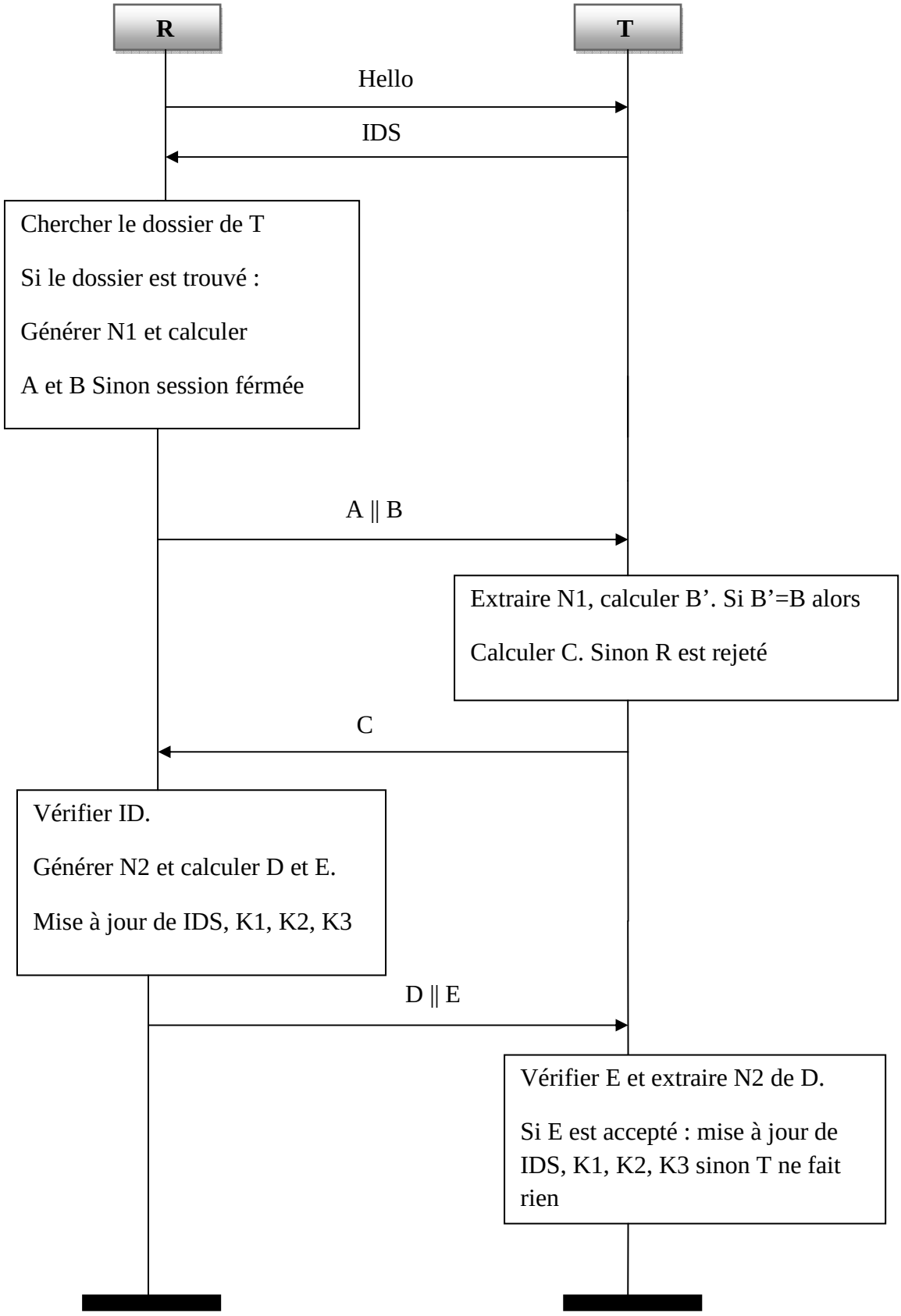


Figure III.2: Diagramme de séquences du protocole R^2AP

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)) $\oplus$ I D. 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:

IDS^{new} = Rec (I DS $\oplus$ N2 || K3) $\oplus$ K1

$K1^{new}$ =Rec (N2 || N1) $\oplus$ K2

$K2^{new}$ = Rec (K2 || N1 $\oplus$ N2) $\oplus$ K3

$K3^{new}$ = 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 proposée pour ce protocole est la suivante (à noter « % » représente un début de commentaire):

R ⟶ T: Hello

T ⟶ R: IDS

R ⟶ T: A || B

T ⟶ R: C

R ⟶ T: D || E % tel que : IDS^{new} = Rec (IDS ⊕ N2 || K3) ⊕ K1

$K1^{new}$ =Rec (N2 || N1) ⊕ K2

$K2^{new}$ = Rec (K2 || N1 ⊕ N2) ⊕ K3

$K3^{new}$ = Rec (K2 || K3) ⊕ N1

III.3.2 Le protocole HMNB

La référence [51] représente le protocole HMNB utilisé pour l'authentification mutuelle, ce dernier utilise la primitive cryptographique qui s'appelle la fonction de hachage. Une attaque d'usurpation et de désynchronisation sont découvertes sur ce protocole [52] [53].

III.3.2.1 Notations

Les notations utilisées dans ce protocole sont présentées dans le tableau III.2:

Notations	Signification
R	Lecteur
T	Tag
ID	Identificateur partagé entre le tag et le lecteur
IDP	L'ancienne valeur de l'ID
HID	Fonction de hachage de l'ID
S	Etat du tag
Nr, Nt	Nombres aléatoires généré par le lecteur et le tag respectivement
||	Concaténation
h	Fonction de hachage

Tableau III.2: Notations utilisées dans HMNB

III.3.2.2 Description du protocole

Ce protocole est basé sur l'authentification mutuelle du tag et du lecteur, il possède quatre étapes comme le montre le diagramme de séquence suivant:

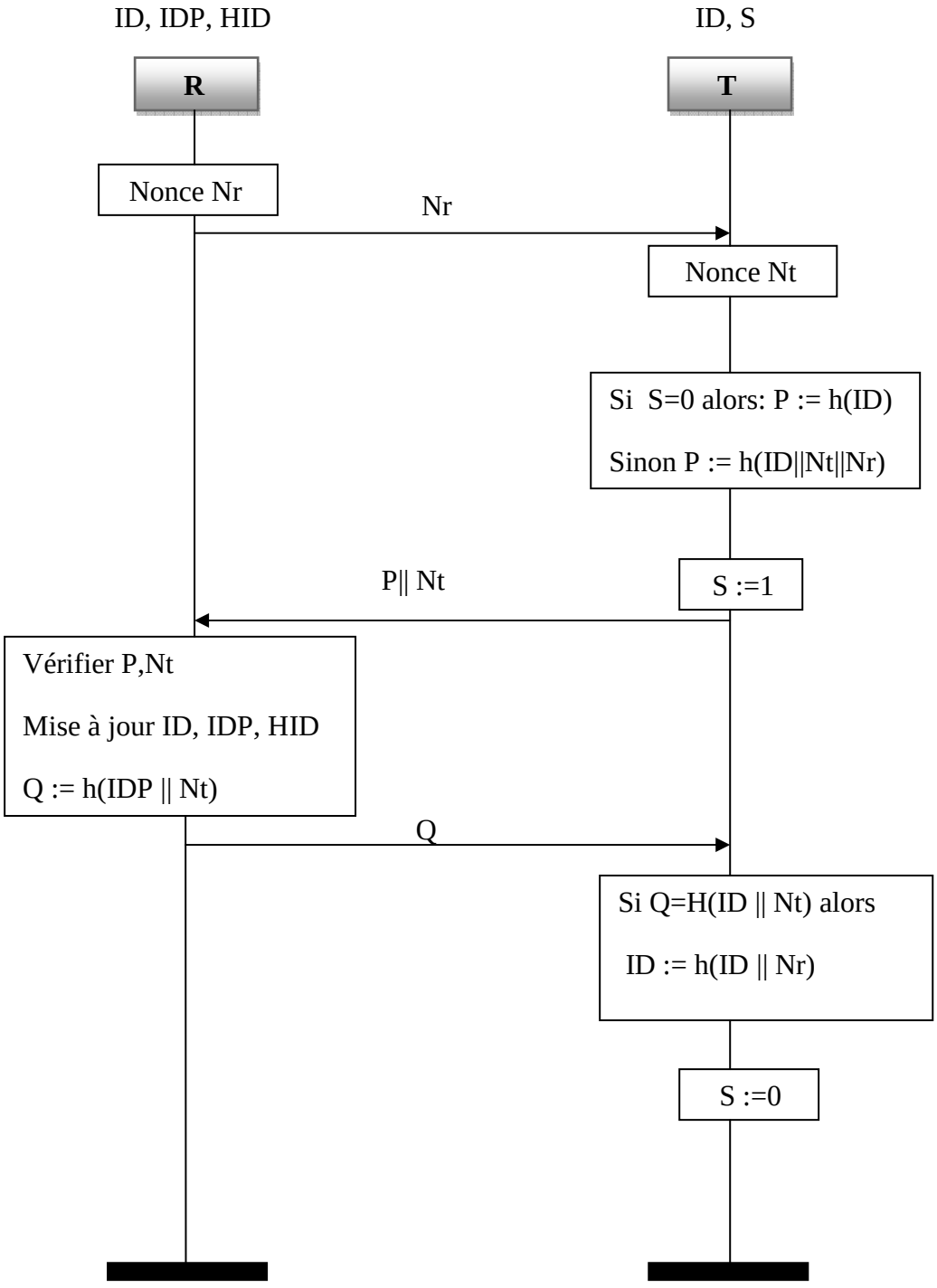


Figure III.3: Diagramme de séquences du protocole HMNB

Étape 1: Le lecteur R génère un nonce aléatoire Nr et l'envoie au tag pour initialiser une nouvelle session de protocole.

Étape 2: T qui reçoit Nr, il génère un nonce aléatoire Nt et vérifié son état, si S=0 il calcule P := h(ID) sinon calcule P := h(ID||Nt||Nr), ce cas devrait se produire rarement dit impossible. Puis met son état à 1 et envoie P accompagné de Nt au R .

Étape 3: À la réception de message envoyé par T, R les utilisent pour vérifier P et Nt, s'ils sont vérifiés R met à jour ID, PID et HID et calcule Q := Q := h(IDP||Nt) et l'envoie à T.

Étape 4: À la réception de message envoyé par R, T l'utilisent pour vérifier si Q= H(ID||Nt), si oui T met à jour son ID puis remet son état à 0.

Les mises à jour se font selon le tableau suivant :

La réponse du tag	Les mises à jour (au niveau du lecteur)
h(ID) || Nt	IDP := ID; ID := h(ID || Nr); HID := h(ID)
h(ID||Nt||Nr) || Nt	IDP := ID; ID := h(ID ||Nr); HID := h(ID)
h(IDP||Nt|| Nr) || Nt	ID := h(IDP||Nr); HID := h(ID)
Autres	Rejeter le tag

Tableau III.3: Mises à jour dans HMNB.

III.3.3 Le protocole de Chien et Huang

Le protocole proposé par Chien et Huang [54], le lecteur R et le tag T partagent leurs clés secrètes k et ID. Le lecteur lance le protocole par l'envoie d'un nonce r1. Le tag génère un nonce aléatoire nt et calcule la fonction de hachage g = h(r1 xor r2 xor ID) et ID2 = rotate (ID,g), le tag envoie la moitié gauche de (ID2 xor g) avec son nonce r2 au lecteur. Le lecteur calcule chaque paire d'ID et k jusqu'à ce que cela trouve le tag correspondant. Il envoie alors la moitié droite de (ID2 xor g) au tag. Une attaque par rejeu algébrique est découverte sur ce protocole [52]**,** sa cause est la fausse utilisation de l'opérateur algébrique xor dans les messages transmis de la fonction g.

III.3.4 Le protocole de Lee et al

Le protocole proposé par Lee et al. [55] basé sur le partage des clés secrètes est utilisé pour l'authentification mutuelle et prévenir la désynchronisation. La valeur initiale de K est stocké dans K'. le lecteur lance le protocole par l'envoi de r0. Le tag génére un nonce r1 et calcule la réponse h(r0 xor r1 xor k) et l'envoi au lecteur avec r1. Le lecteur utilise la réponse de tag pour chercher K dans la base de donnée puis calcule h(h(r0 xor r1 xor k) xor k xor r0) et l'envoi au tag. Une attaque sur l'authentification du tag est découverte sur ce protocole [52].

III.3.5 Le protocole LMAP (Lightweight Mutual Authentication Protocol)

LMAP [56] appartient à la famille des protocoles RFID ultraléger appelés UMAP (Ultralightweight Mutual Authentication Protocol), proposés par Lopez et al on 2006. Ce protocole est utilisé pour l'authentification mutuelle entre le tag et le lecteur, il utilise des fonctions simples comme xor, addition… ect. Le lecteur lance le protocole en envoyons le message ''Hello'' au tag, le tag répond avec son IDS. R après la réception de IDS, il génère deux nombres aléatoire R1 et R2 puis calcule A, B et C avec R1 et R2 et les clés symétriques K1, K2, K3 puis les envoie au tag. Le tag après la réception de A,B et C il extrait R1 et R2 pour calculer D puis met à jour ses clés symétriques K1,K2,K3,K4 et son IDS et envoie D au lecteur qui lui aussi met à jour ses valeur secrets après la réception de message D. Une attaque de désynchronisation et divulgation ont été découvertes sur ce protocole [41]

III.4 Conclusion :

Les protocoles cryptographiques sont un moyen important pour assurer la sécurité des communications entre les composants des systèmes RFID, pour cela ces protocoles doivent être vérifiés avant leurs mises en services.

Dans ce chapitre nous avons présentés les types des protocoles cryptographiques, et la description de quelques protocoles cryptographiques.

Le chapitre suivant prend la charge de définir la vérification formelle toute on s'intéressant à ses outils automatisés.

CHAPITRE IV : La vérification formelle et l'outil AVISPA.

IV.1 Introduction

De nos jours, sur plusieurs protocoles cryptographiques publiés, 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.

Nous énonçons donc dans ce chapitre les principales propriétés concernant les protocoles cryptographiques, ainsi 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. Nous mettons l'accent sur les principaux aspects de modélisation adoptés par cet outil.

IV.2 Les notions de base de la vérification

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

IV.2.1 Les protocoles

Les protocoles de sécurité visent alors, comme leur nom l'indique, à établir entre deux ou plusieurs participants (agents), des communications répondant à certaines propriétés de sécurité telles que la confidentialité, l'intégrité, l'authentification. Il s'agit de petits programmes qui spécifient une séquence d'émissions/réceptions de messages, en précisant la forme et le contenu de ces derniers, l'ordre de cette séquence d'échanges, ainsi que les participants impliqués [28].

IV.2.2 Les agents

Chaque agent (participant) peut effectuer, simultanément, avec différents participants, plusieurs sessions du protocole. Les participants sont de deux types [57]:

- ***Participants honnêtes***

Dans la description d'un protocole, les agents sont supposés être honnêtes, *i.e.* ils ont un comportement qui suit celui énoncé par une exécution normale du protocole. Les envois de messages sont précis et effectués dans l'ordre défini par le protocole. De plus, ces agents honnêtes ne collaborent pas avec les attaquants.

- ***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.

IV.2.3 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 [57] [26].

IV.2.4 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.

Montrer que une attaque ou un intrus obtient le secret échangé entre les agents honnêtes prouve que le protocole ne vérifie pas la propriété de secret, alors le message est accessible pour l'intrus. Il existe donc deux approches complémentaires pour vérifier la propriété de secret : soit nous cherchons une preuve que le secret est préservé, soit nous recherchons une attaque [26].

IV.3 Les objectifs de la vérification

La vérification de la sécurité des protocoles cryptographiques s'articule autour de deux axes complémentaires : rechercher une attaque ou prouver que le protocole est sûr [57].

IV.3.1 Trouver une attaque

Le premier axe de la vérification consiste à chercher une attaque sur un protocole. Observons que la majeure partie des attaques sur internet sont découvertes par des personnes malicieuses. La recherche d'attaques n'est pas une tache aisée même pour un spécialiste, car l'esprit humain est faillible. Ainsi, de nombreux outils de vérification automatique de protocoles ont vu le jour et constituent une aide précieuse à la découverte d'attaques comme par exemple l'outil AVISPA, CASPER/FDR, COPROVE, Proverif.

IV.3.2 La sûreté d'un protocole

Un autre point important de la vérification est de prouver qu'il n'existe pas d'attaque sur un protocole donné en présence d'un intrus possédant certaines capacités, ce qui manuellement, peut être fastidieux et complexe. Certains outils prouvent la propriété de sûreté d'un protocole, il est possible de vérifier formellement si un protocole est sûr contre les attaques menées par un intrus possédant certaines capacités. Si cette vérification formelle échoue, elle nous fournit dans le cas d'une abstraction correcte une preuve de la non-sûreté du protocole, cette preuve correspond effectivement à une véritable attaque si la vérification est complète.

IV.4 Les modèles de vérification

Il existe deux principales méthodes de vérification, la vérification formelle (approche symbolique) et de l'approche calculatoire [58]:

IV.4.1 La vérification calculatoire

Cette approche permet de raisonner sur des questions de complexité et de probabilité, car les primitives cryptographiques sont considérées comme de vrais algorithmes et l'intrus y est modélisé comme un algorithme qui peut s'exécuter efficacement. Le fait qu'un protocole satisfait une propriété dans ce modèle garantit qu'il n'existe pas d'algorithme efficace pour la violer. Ce modèle permet donc d'obtenir des résultats forts. Le point faible consiste en le fait que les preuves sont souvent pénibles et se prêtent mal à une automatisation [28] [26].

Les tâches de vérification des contraintes de sécurité font apparaitre le besoin de prouver que ces processus de vérification sont fiables. Puisque l'informatique est basée sur les mathématiques, les preuves du fonctionnement des programmes sont présentées par des formules mathématiques. Cette formalisation des processus, par des formules et des algorithmes mathématiques, nous emmène dans un domaine que nous appelons les méthodes formelles.

IV.4.2 La vérification formelle

La vérification formelle est une combinaison de modèle mathématique d'un protocole de sécurité, et les propriétés souhaitées, ainsi qu'une procédure efficace pour déterminer s'il existe des preuves que le protocole satisfait ces propriétés [58].

Pour commencer l'étude des protocoles cryptographiques il est nécessaire de les modéliser, ensuite, une vérification formelle à l'aide de cette modélisation permet de conclure que le protocole ne révèle en aucun cas une certaine faille ou qu'il préserve une certaine propriété. Les conclusions de la vérification permettent de détecter des attaques ou, au moins, des points faibles que la spécification d'un protocole peut comporter.

Le point fort d'un modèle symbolique est le fait qu'il permet d'obtenir des preuves mathématiques simplifiées et souvent automatisées.

Nous décrivons brièvement 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 [27] [28].

- ***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 transitions***

Pour donner une sémantique à la définition d'un protocole, nous utilisons des systèmes de

transitions. Dans un tel système, 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 canal de communication sert à l'échange des messages entre les agents. Il est possible de distinguer deux types de canaux de communications : les canaux de communications publics et les canaux de communications privés [57].

Un canal de communication est dit public si tous les messages émis sur ce canal peuvent être vus par tous les participants, qu'ils soient honnêtes ou pas. Les messages échangés sur ces canaux sont donc 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 communications privés sont des canaux de communications définis entre certains participants honnêtes, et seuls ces participants peuvent recevoir et envoyer des messages sur ces canaux. Par conséquent, un intrus ne peut donc pas écouter les messages qui circulent sur ce genre de canaux.

IV.5 Les méthodes de vérification formelle

On distingue 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].

IV.5.1 La méthode du raisonnement logique

C'est la première des méthodes utilisées en matière de la vérification automatique des protocoles. On peut citer trois techniques essentielles :

- ***La BAN*** : C'est la plus importante technique de raisonnement, dont les auteurs sont Mike **B**URROWS, Martin **A**BADI, Roger **N**EEDHAM. 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 [60].

- ***La GNY (extension de la BAN)*** : Dont les auteurs sont **G**ong, **N**eedham et **Y**ahalom. Plus sophistiquée et complexe, elle améliore la BAN en mettant en évidence la différence entre le contenu d'un message et sa signification [61].

- ***La BGNY (extension du GNY)*** : de **B**ackin, c'est une 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…) [61].

IV.5.2 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 [57].

IV.5.3 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 [62].

Les méthodes d'exploration d'Etat 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.6 Le modèle d'intrus de Dolev-Yao

Le modèle d'intrus Dolev-Yao est le premier modèle d'adversaire formel proposé par Dolev et Yao en 1981, destiner à la vérification de protocoles cryptographiques, et représente les capacités que possède un attaquant pour acquérir de l'information. Cette formalisation idéalise les protocoles en considérant les points suivants [57] [63] :

IV.6.1 Un réseau idéal

Le modèle de Dolev-Yao ne prend pas en considération la notion de temps, ni les pertes de messages, il suppose un réseau idéal c'est-à-dire qu'aucun message échangé n'est perdu et tous les messages sont envoyés et reçus instantanément par les agents. Ainsi, avec le modèle introduit par Dolev et Yao, il n'est pas possible de prendre en compte le temps dans les capacités de l'attaquant. De plus, dans ce modèle, tous les canaux de communications sont supposés être publics, ainsi nous sommes dans un environnement le plus hostile possible pour effectuer la vérification.

IV.6.2 Abstraction des messages échangés

En réalité, les messages échangés par des agents lors d'une communication sont des chaînes de bits. Dans ce modèle, comme dans de nombreuses approches symboliques en vérification de protocoles, les messages échangés sont abstraits par des termes générés par une algèbre de termes qui contiennent les symboles des fonctions de chiffrement, de hachage, un ensemble de messages.

IV.6.3 Chiffrement parfait

Dans ce modèle on fait l'hypothèse de cryptographie parfaite, ainsi les messages chiffrés sont donc analysés comme des boites noires contenant un message qu'il n'est possible d'ouvrir que si la clé de déchiffrement est connue.

IV.6.4 Capacités de l'intrus

Dans le modèle de Dolev et Yao, un intrus peut utiliser n'importe quel terme qu'il a observé sur le réseau autant de fois qu'il le désire. Ainsi, les messages qui circulent sur le réseau, sont connus par l'intrus et constituent l'ensemble de sa connaissance initiale.

Donc, dans ce système, on considère que l'intrus a le contrôle total du réseau. Il connait toutes les données publiques des agents, dispose des privilèges et des clés des agents malhonnêtes. Ainsi, l'intrus peut appliquer une des règles données par le modèle : Il peut déchiffrer un message s'il connaît la clé de déchiffrement, il peut chiffrer un message avec n'importe quelle clé en sa possession et il est capable de mémoriser, effacer, construire et envoyer tous les messages s'il a la clé.

Dolev et Yao ont modélisé les capacités de l'intrus par un modèle de déduction c'est à dire qui représente tous les messages qu'un intrus peut fabriquer par un ensemble de règles.

Pour montrer qu'un protocole respecte une propriété donnée, il faut fournir une preuve formelle que quelles que soient les actions de l'attaquant, la propriété désirée est maintenue pour le protocole.

IV.7 Les outils de vérification formelle

Il existe des outils automatiques qui vérifient les protocoles cryptographiques en présentons brièvement certains [66] [67] [68] :

IV.7.1 L'outil AVISPA

Cette plateforme comprend quatre modules d'analyse qui implémentent différents algorithmes de verifcation, ce qui permet d'avoir une étude plus complète: On-the-Fly-Model Checker(OFMC), The constraint-Logic-based AttackSearcher (CL-AtSe), SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximation for the Analysis of Security Protocols (TA4SP) [64].

Récemment, la propriété du << ou exclusif >> à été ajoutée a l'outil CL-ATSE. Cet outil essaye dans sa dernière version d'adopter une approche modulaire afin de prendre en compte différentes théories équationnelles. Cet outil sera détaillé dans la prochaine partie.

IV.7.2 L'outil ProVerif

Développer par B. Blanchet [65] est un vérificateur automatique de protocoles cryptographiques qui peut traiter beaucoup de primitives cryptographiques, dont la cryptographie à clé partagée (symétrique) et publique (chiffrement et signatures), les fonctions de hachage automatiquement convertie en clauses de Horn. Ce vérificateur de

protocoles permet de vérifier des propriétés de secret et aussi des propriétés d'authentification et peut traiter un nombre non-borné de sessions.

IV.7.3 L'outil Hermes

Est un outil dédié à la vérification de propriétés de secret pour les protocoles cryptographiques pour un nombre non borné de session. Permet de vérifier si une donnée est secrète dans un protocole, il génère l'ensemble des données qu'un intrus ne doit pas apprendre et l'ensemble de données non secrètes. Un des objectifs de la nouvelle version de cet outil serait de prendre en compte la notion de temps pour la vérification des protocoles cryptographiques [66].

IV.7.4 L'outil Casper

Développer par G. Lowe et A.W. Roscoe [67], modélise partiellement le chiffrement de Vernam (qui utilise le ≪ ou exclusif ≫), Grace à cet outil G. Lowe trouva certaines attaques sur le protocole TMN, de plus cet outil permet d'analyser les protocoles avec timestamps sous certaines hypothèses.

IV.8 Présentation de l'outil AVISPA :

AVISPA (Automated Validation of Internet Security Protocols and Applications) a été développé en 2004 par Basin et Al dans le cadre d'un projet européen. C'est un outil d'analyse automatique destiné à aider à la validation de protocoles. Il possède deux enjeux majeurs : être performant, tout en garantissant l'accessibilité aux non-spécialistes du domaine [60]. L'outil est utilisable sur des protocoles à petite et moyenne échelle (tels fournis dans la librairie de Clark/Jacob[1]), ainsi que des protocoles de sécurité Internet à grande échelle [59].

IV.8.1 Architecture logicielle de l'outil AVISPA (back-end AVISPA) :

AVISPA dispose de quatre différents back end comme illustré sur la Figure IV.1 et qui implémente des techniques d'analyse allant de la falsification de protocole (découverte d'attaque sur le protocole d'entrée) à des méthodes de vérification à base d'abstractions pour des nombres fini ou infini de sessions. Si une propriété de sécurité est violée dans la spécification, l'analyseur produit la trace de séquence d'événement qui mène à une faille et affiche quelle propriété est violée.

[1] La librairie de Clark/Jacob est une librairie de spécification informelle aisément compréhensible employée dans [Clark and Jacob, 1997] pour décrire les protocoles cryptographiques.

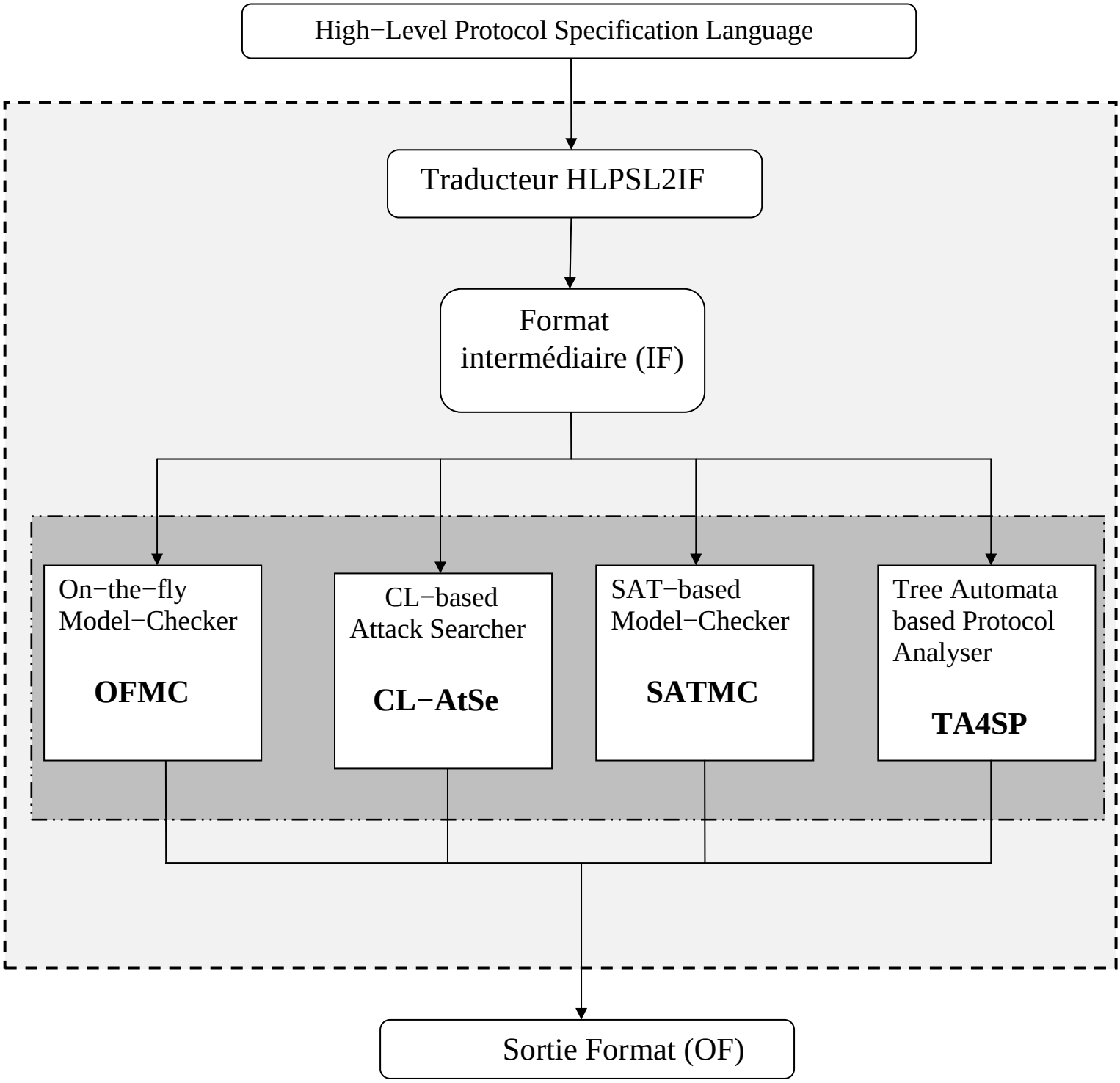


Figure IV.1: Architecture logicielle d'AVISPA [64] [70]

- ***OFMC (On-the-Fly Model Checking) :***

Historiquement, l'histoire d'OFMC a commencé au sein du projet AVISS, et puis OFMC a mûri au sein du projet AVISPA. Il effectue une vérification bornée en explorant le système de transition décrit par une spécification IF. OFMC implémente des techniques symboliques correctes et également complètes. Il supporte la spécification des opérateurs à propriétés algébriques tels que le OU exclusif ou encore l'Exponentielle [70]. Il est utilisé pour un protocole dans lequel les propriétés algébriques des fonctions cryptographiques sont importantes [59].

- ***CL-AtSe (Constraint logic - based Attack Searcher)***

C**'**est un outil basé sur des techniques de résolution de contraintes et implémente une procédure de décision. Il permet de faire une traduction d'une spécification d'un protocole de sécurité sous forme de relations de transition au format IF, vers un ensemble de contraintes qui peuvent être utilisées pour trouver des attaques sur le protocole en question. Les deux méthodes (la traduction et la vérification) sont totalement automatiques et prises en charge par l'outil CL-AtSe sans intervention d'outils externes. Les possibilités de CL–AtSe ont été étendues lors du projet AVISPA pour supporter des opérateurs possédant des propriétés algébriques (Xor ou Exp) [60]. Il est basé sur les contraintes. Il procède à une simplification du protocole d'origine en éliminant les états redondants par application de propriétés heuristiques. Sa modularité permet d'intégrer facilement d'autres spécifications de fonctions cryptographiques [59].

- ***SATMC (SAT-based Model-Checker)***

Cet outil a été développé au laboratoire DIST à Gènes (Italie). Il construit une formule propositionnelle codant un déploiement borné du système de transition de IF, l'état initial et l'ensemble des états représentant la violation des propriétés de sûreté spécifiées en IF (a fortiori en HLPSL). La formule propositionnelle est ensuite donnée à résoudre à un solveur SAT, sélectionné parmi ces quatre : zCHAFF, mCHAFF, SIM et SATO. Ensuite, tout modèle satisfaisant cette formule est retourné sous forme d'attaque [60]. Il utilise un état transitoire et recherche les éventuelles violations d'un protocole donné. Ceci lui permet de générer une formule représentant la violation et la transforme en attaque [59].

- ***TA4SP (Tree-Automata-based Protocol Analyzer)***

La particularité de cet outil de vérification est qu'à partir d'un état initial il fait soit une sur approximation ou une sous approximation des connaissances de l'intrus en utilisant des automates d'arbres. Cette méthode permet de savoir si un certain état est accessible ou non et que l'intrus peut savoir certaines connaissance ou non et ainsi de conclure l'absence d'attaque sur le secret pour des scénarios exécutés un nombre indéterminé de fois [60]. Autrement dit, il montre la vulnérabilité d'un protocole ou la prédit en faisant une estimation précise des capacités de l'intrus.

IV. 8.2 La démarche AVISPA

AVISPA est une plate-forme qui permet l'analyse et la validation de manière automatique des protocoles de sécurité. Une telle validation repose sur la spécification des protocoles de sécurité à l'aide d'un langage d'entrée appelé HLPSL (High Level Protocol Specification Language). Ces spécifications sont ensuite traduites en un format de plus bas niveau appelé format intermédiaire (Intermediate Format IF) à l'aide d'un traducteur appelé hlpsl2if

Translator. Le résultat de cette traduction est un fichier .if à partir duquel différents outils effectuent la vérification.

AVISPA couvre un large spectre de protocoles et combine plusieurs techniques de vérification qui reposent soit sur la vérification de modèles, soit sur la résolution de contraintes d'exploration ''symbolique" des protocoles. Les différents back-end de plateforme AVISPA sont dédiés à la détection d'attaques. Un protocole est dit non sécurisé par rapport à une propriété ou que l'objectif de sécurité dans la spécification d'un protocole est violé lorsqu'on a la possibilité d'atteindre des états où, les mots de passe apparaissent en clair ou que l'intrus arrive à décrypter un message avec les informations qu'il a, ou usurper l'identité d'un autre agent. Dans ces cas, les back-end de cet outil fournissent une trace qui montre la séquence des événements qui ont conduit à cette violation.

IV.8.3 L'outil graphique SPAN

Le rôle de SPAN (Security Animator for AVISPA) est d'aider à la conception des spécifications formelles. SPAN permet d'animer les spécifications HLPSL, i.e. de produire interactivement des MSC (Message Sequence Charts) qui peuvent être vus comme une trace "Alice & Bob" d'une spécification HLPSL [71]. Il y a deux modes de simulation:

a. En mode normal, les seules transitions qu'on voit sont celles où l'émetteur et le récepteur du message sont d'accord sur le message.

b. En mode intrus, tous les messages peuvent être envoyés à l'intrus et chaque message qu'il reçoit enrichit ses connaissances.

SPAN a plusieurs avantages dont celle de vérifier que la spécification HLPSL correspond bien au protocole qu'on avait en tête avant même de chercher des attaques. On peut ainsi trouver des erreurs de spécification que les outils de vérification ne détecteraient pas ou n'expliciteraient pas correctement. SPAN peut même être un grand secours pendant la phase de conception du protocole. Plutôt que de dessiner difficilement des schémas incomplets ou mal expliqués, il peut être très efficace d'animer avec SPAN les premières idées que l'on a et de faire des tests dès le début de la conception. Cependant, le plus grand intérêt de SPAN est la possibilité de reconstruire les attaques fournies par les outils de vérification, ou même de les construire à partir d'une intuition. SPAN permet à l'utilisateur de choisir à chaque étape si les messages sont reçus par l'intrus ou non et affiche dans un cadre les connaissances de l'intrus [59] [71].

IV.8.4 Les étapes de vérification

1- On commence par la spécification du protocole à tester grâce au langage HLPSL, ainsi que les propriétés à vérifier.

2- On lance AVISPA à l'aide de l'interface graphique SPAN tout en précisant le fichier .hlpsl et l'analyseur (back end) qu'on va utiliser et par défaut AVISPA utilise OFMC back end.

3- Ensuite AVISPA, après l'analyse, il déclare que soit le protocole est sûr (mais peut être sous certaines conditions), ou bien le protocole présente des failles. Ou bien il génère des

erreurs de compilation, et dans ce cas, on pourra changer la spécification du protocole et poursuivre depuis l'étape 2.

IV.9 Présentation du langage de spécification des protocoles HLPSL

Pour écrire des spécifications de protocoles de sécurité, AVISPA dispose d'un langage appelé HLPSL afin de spécifier ces protocoles. HLPSL est un langage de spécification de protocoles de sécurité inspiré des travaux de Lamport à propos de la logique temporelle des actions TLA (Temporal Logic of Actions) [59] [70]. La représentation des protocoles de sécurité est basée sur des systèmes d'états/transitions sur lesquels sera effectuée la vérification des propriétés de sûreté qui sont exprimées en logique temporelle linéaire.

De plus, le domaine de l'analyse des protocoles fait appel à plusieurs constructions syntaxiques (tels que la structure du message) et à des concepts sémantiques (comme la notion de l'intrus). Idéalement, il serait pratique de modéliser des protocoles dans une langue qui offre ces points communs intégré. Le développement de HLPSL a été donc effectué avec les objectifs de conception suivants:

a. Il doit fournir un moyen pratique, lisible par l'homme, et un langage facile à utiliser et être encore assez puissant pour supporter la spécification des protocoles Internet modernes. Pour ça, HLPSL a été défini de manière à ressembler formellement le langage de définition de transitions gardées dans un système de transitions d'états et est équipé des constructions qui permettent la spécification modulaire de protocoles.

b. Il doit disposer d'une sémantique formelle. A cet effet, HLPSL a été basée sur ce Lamport TLA et sa sémantique qui est donnée par une traduction à un sous - ensemble de TLA.

c. Il doit se prêter à une analyse formelle automatisée. Ceci est réalisé par une translation de HLPSL dans le format intermédiaire (IF).

Dans les spécifications HLPSL, sont représentées toutes les entités qui peuvent intervenir dans le protocole en question. Cette description est faite dans un fichier qui a pour extension .hlpsl et qui fera l'objet d'une traduction à l'aide du traducteur hlpsl2if afin de générer un fichier .if. Ensuite, les différents back-end utilisent ce format pour vérifier le protocole spécifié.

IV.9.1 Structure d'une spécification HLPSL

Une spécification HLPSL est composée de trois parties : Une liste de définition de rôles, une liste des objectifs ou des propriétés de sûreté à vérifier, et enfin, l'instanciation du rôle principal (généralement sans arguments) [60] [64].

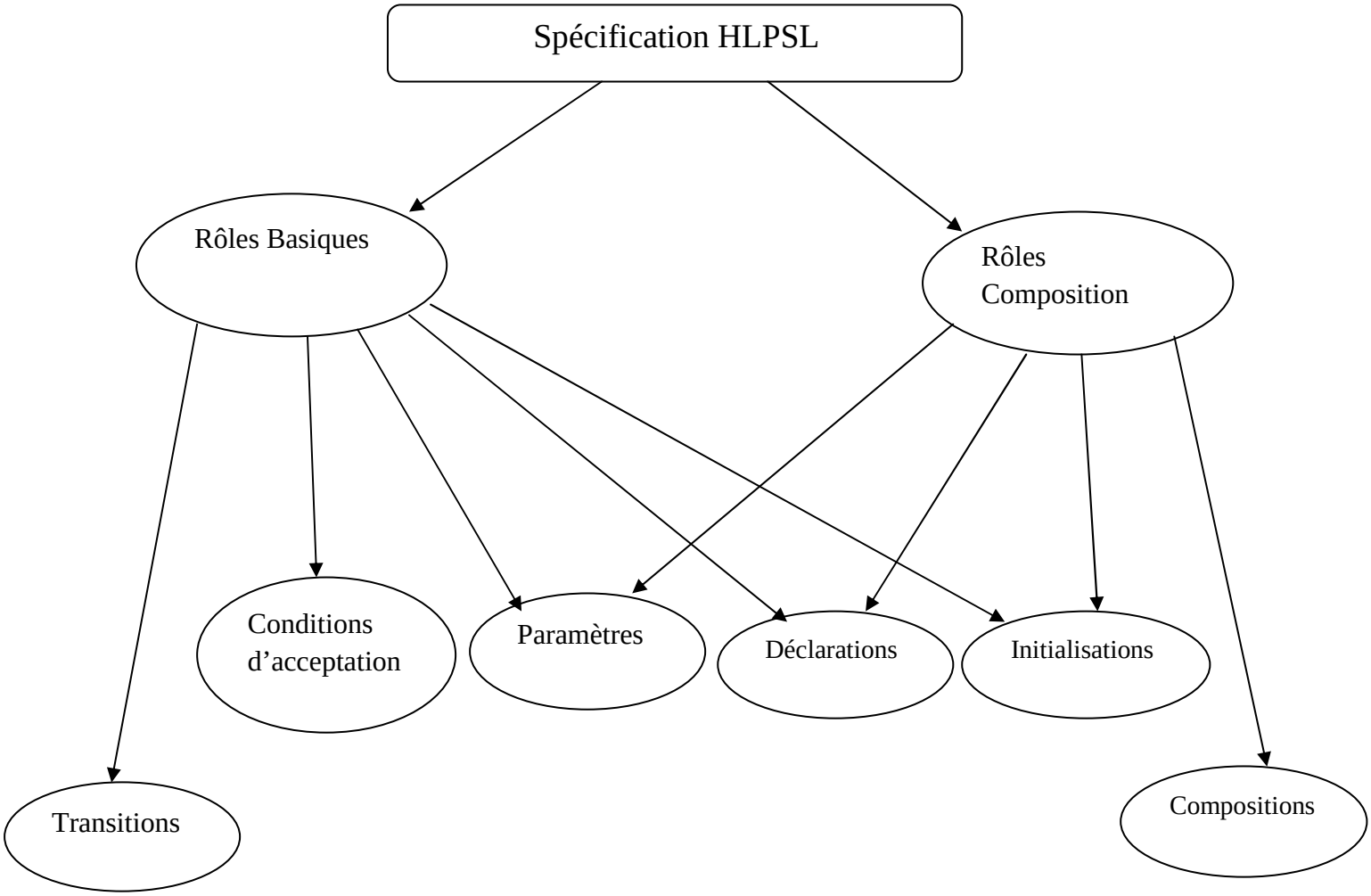


Figure IV.2 : Structure d'une spécification HLPSL

IV.9.1.1 Définition des rôles :

Un rôle HLPSL peut être considéré comme une description du comportement. Les rôles peuvent être paramétrés en prenant un ou plusieurs arguments de types arbitraires et peut également déclarer des variables locales par l'intermédiaire des opérateurs existant. On distingue deux types de rôles différents:

- ***Les rôles basiques****:* Ils représentent la connaissance initiale et le comportement de chaque participant (agent) intervenant dans le protocole spécifié.

La connaissance initiale liée à chaque rôle (soit Alice au Bob) est exprimée par une liste de paramètres qui sont : la clé symétrique, les différents agents connus par ce rôle, un ensemble de canaux qui seront utilisés pour la communication… etc. Dans la déclaration d'un rôle, on peut trouver une clause optionnelle ''played_by'' pour spécifier quel agent joue le rôle

considéré. Nous pouvons trouver aussi une section qui contient une liste de variables locales et leurs types, qui peuvent être initialisées dans la section ''init''. Une liste de transitions est définie dans la dernière partie du rôle. Cette dernière va définir le comportement de l'agent à travers des transitions qu'il peut faire suivant une certaine condition pour changer d'état.

- ***Les rôles de composition:*** Ils servent à définir les scénarios des rôles basiques (i.e. cette catégorie permet l'instanciation des différents rôles basiques pour modéliser la totalité du protocole) et permet de combiner les autres rôles soit en parallèle ou en séquence, et définir ainsi le protocole en tant que session. Chaque spécification doit avoir un rôle spécial qui peut s'appeler par exemple session et qui va représenter une session du protocole.

Il existe un autre rôle de composition couramment appelé environnement ou rôle principal**.** Ce rôle ne possède aucun paramètre et sert à définir l'état initial du système en précisant la connaissance initiale de l'intrus par la clause intruder_knowledge**,** ainsi qu'un nombre fini de rôles session.

IV.9.1.2 Déclaration des propriétés de sûreté

Les propriétés de sûreté du protocole à évaluer par AVISPA telles que le secret, l'authentification sont déclarées dans une section à part et afin de les vérifier, il est nécessaire d'introduire dans la spécification des éléments événementiels appelés signaux. En HLPSL, il existe différents signaux permettant d'exprimer les propriétés de sûreté:

- **secret(E, id, S) :** déclare que l'information E est un secret partagé par un ensemble S d'agent.

- **witness(A, B, id, E) :** pour la propriété d'authentification (faible) de l'agent A auprès de B grâce à la donnée E. Cet objectif sera identifié par la constante id dans la section réservée à la déclaration des propriétés de sécurité.

- **request (B, A, id, E) :** pour l'authentification forte entre A et B. Elle déclare que B demande une vérification de la valeur E. La fonction de id est la même que pour witness.

- **wrequest(B, A, id, E) :** elle est similaire à **request()**, mais cette fois pour l'authentification faible.

La déclaration des objectifs ou des propriétés à vérifier se fait dans une section à part (goals). Chaque propriété est identifiée par une constante qui réfère le prédicat défini (secret, witness, request et wrequest) pour une transition donnée.

IV.9.1.3 Instanciation d'un rôle

La création d'une instance d'un rôle est comme l'appel d'une procédure, en donnant une valeur pour chaque paramètre déclaré dans le rôle, et bien sûr le nombre d'arguments ainsi que leurs types doit être les mêmes que ceux des paramètres formels du rôle. Pour le rôle principal, il suffit d'invoquer son nom (usuellement sans paramètre comme suit : environnement()).

IV.9.1.4 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 role Alice et role Bob

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: role session et role %envirenment

role session (arguments)

} Listes des déclarations

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

role environment( )

} Liste des declarations et les connaissances initiales de l'intru

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

end role

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

goal

} Les propriétés à vérifier

end goal
environment()

IV.9.2 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.

- flux de contrôle assurant les transitions valides.
- propriétés de sécurité à vérifier : confidentialité, authentification, intégrité.

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

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

IV.10 Conclusion:

La vérification formelle et automatique des protocoles de sécurité est une méthode qui permet d'assurer la fiabilité des protocoles cryptographiques.

Dans ce chapitre nous avons présenté quelques des notions importantes pour la vérification, et les caractéristiques de la vérification formelle et ses méthodes ainsi, un ensemble d'outils qui permet d'assurer cette vérification, nous avons basé sur l'outil AVISPA, le langage HLPSL et nous avons cerné quelques problèmes liés à ce dernier.

Dans le chapitre suivant nous allons vérifier automatiquement les protocoles R2AP, HMNB et en présentant une amélioration à ce dernier.

CHAPITRE V : Vérification formelle d'un protocole RFID à l'aide d'AVISPA.

V.1 Introduction

La vérification des protocoles de sécurité est vite devenue incontournable au vu des actions malveillantes, de tentatives d'intrusion et d'attaques diverses que subissent les réseaux informatiques. Comme vu dans les chapitres précédents, les failles dans les protocoles cryptographiques exploitent essentiellement des erreurs logiques. C'est pourquoi depuis les années 90 des outils de vérifications automatiques comme AVISPA sont des moyens efficaces et performants pour vérifier la robustesse des protocoles cryptographiques.

Dans ce chapitre on va vérifier les propriétés de sécurité de deux protocoles de sécurité dans les systèmes RFID à l'aide de l'outil AVISPA, et on va présenter une amélioration pour l'un de ses protocoles.

V.2 Les vérifications sur AVISPA & SPAN

Dans notre travail on a utilisé la version exécutable de la plate forme AVISPA & SPAN téléchargeable sur le site : www.avispa-project.org .Après, la spécification du protocole en HLPSL, accéder au répertoire et charger un fichier avec l'extension .hlpsl contenant la spécification en HLPSL, puis transformer automatiquement ce fichier en une description du protocole au format IF (Format Intermédiaire). Ce fichier qui est au format intermédiaire sera automatiquement envoyé comme une entrée aux quatre outils qui vont vérifier le protocole et présenter leurs diagnostics dans un format de sortie commun. Ces diagnostics fournissent dans le cas échéant, une trace d'attaque qui peut être visualisée graphiquement.

V.3 La vérification formelle du protocole R^2AP

Dans cette section, on vérifie les propriétés de la confidentialité de l'identificateur ID et l'index IDS (sec_ID et sec_IDS respectivement), l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader) du protocole R2AP décrit dans le chapitre III. Ces propriétés sont spécifiées dans HLPSL au niveau de la partie goal comme suit:

```
goal
secrecy_of sec_ID, sec_IDS      % confidentialité de l'ID et IDS
authentication_on aut_tag       % autentification du tag par C
authentication_on aut_reader    % authentification du lecteur par A et B
end goal
```

- ***Notations***

Le tableau suivant présente les notations utilisées dans la spécification HLPSL de R^2AP :

Notations	Signification
r	Lecteur légitime
t1, t2	Tag légitimes
idt1, idst1	Identificateur et index du tag t1 respectivement
idt2, idst2	Identificateur et index du tag t2 respectivement
kt11, k12t, kt13	Clés symétriques partagées entre r et t1
kt21, kt22, kt23	Clés symétriques partagées entre r et t2
idi, idsi	Identificateur et index de l'intru respectivement
k1ir, k2ir, k3ir	Clés symétriques partagées entre le lecteur légitime r et l'intru i
k1it, k2it, k3it	Clés symétriques partagées entre le tag légitime t et l'intru i

Tableau V.1 : Notations utilisées dans la spécification HLPSL de R^2AP

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 :*

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,
idt1,idst1,idt2,idst2: text,
rot,rec:hash_func,
kt11,k12t,kt13,kt21,kt22,kt23:symmetric_key,
aut_tag, aut_reader:protocol_id
intruder_knowledge = {t1,t2,r,rot,rec}
composition

% attaque replay
session(t1,r,idt1,idst1,rot,rec,kt11,k12t,kt13)   /\
session(t2,r,idt2,idst2,rot,rec,kt21,kt22,kt23)
```

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\R2AP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 12 nodes
  depth: 4 plies
```

Ce résultat signifie en clair qu'il n y a pas d'attaque par rejeu, Cela confirme l'exactitude de ce qu'ils ont dit les auteurs de [50].

- ***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 t,r : agent,
id,ids,idi,idsi: text,
rot,rec:hash_func,
k1,k2,k3,k1it,k2it,k3it,k1ir,k2ir,k3ir : symmetric_key,
aut_tag, aut_reader:protocol_id
intruder_knowledge = {t,r,rot,rec,idi,idsi,k1it,k2it,k3it,k1ir,k2ir,k3ir}
composition

%attaque main in-the-middel
session(t,r,id,ids,rot,rec,k1,k2,k3)
/\ session(t,i,idi,idsi,rot,rec,k1it,k2it,k3it)
/\ session(i,r,idi,idsi,rot,rec,k1ir,k2ir,k3ir)
end role
```

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

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 C:\progra~1\SPAN\testsuite\results\R2AP.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed  : 0 states
 Reachable  : 0 states
 Translation: 0.01 seconds
 Computation: 0.00 seconds
```

D'après ce résultat on peut déduire que ce protocole est résistant à l'attaque main in-the-middle comme l'ont souligné les auteurs de [50].

V.4 La vérification formelle du protocole HMNB

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

```
goal
    secrecy_of sec_ID, sec_IDP    % confidentialité de l'ID et IDP
    authentication_on aut_reader  % authentification du lecteur par Q
    authentication_on aut_tag     % authentification du tag par HID
end goal
```

- ***Notations***

Les notations utilisées dans la spécification HLPSL du protocole HMNB sont présentées dans le Tableau V.2:

Notations	Signification
r	Lecteur légitime
t	Tag légitimes
idi	Identificateur de l'intru

Tableau V.2 : Notations utilisées dans la spécification HLPSL de HMNB

Comme nous avons cité dans la section précédente, il y a deux attaques d'authentification possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle.

- ***Attaque par rejeu***

La spécification suivante du rôle environnement représente une session entre un tag légitime et un lecteur légitime (t et r). Ce scenario permet de détecter les attaques du type "Attaque par rejeu" s'elles existent.

```
role environment() def=
const t,r : agent, id: text, h: hash_func,
aut_tag, aut_reader: protocol_id
intruder_knowledge = {t,r,h}
composition

%Attaque par rejeu
session(t,r,id,h)
end role
```

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

```
% OFMC
% Version of 2006/02/13
SUMMARY
 UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\HMNB .if
GOAL
 authentication_on_aut_tag
```

Le protocole HMNB est vulnérable et l'attaque détecté est au niveau de but d'authentification du tag

```
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 3 nodes
  depth: 2 plies
ATTACK TRACE
i -> (r,3): start
(r,3) -> i: Nr(1)
i -> (t,3): x240
(t,3) -> i: h(id).Nt(2)
 i -> (r,3): h(id).x250
(r,3) -> i: h(id.x250)
```

- ***Attaque main in-the-middle***

La spécification suivante du rôle environnement représente trois sessions entre tag, lecteur et intrus (t, r et i). Ce scenario permet de détecter les attaques du type "Attaque main in-the-middle" s'elles existent.

```
role environment() def=
const t,r : agent, id,idit,idir : text, h: hash_func,
aut_tag, aut_reader: protocol_id
intruder_knowledge = {t,r,idi,h}
composition
session(t,r,id,h)  /\ session(t,i,idi,h) /\ session(i,r,idi,h)
end role
```

Après la vérification de ce protocole par l'outil CL-Atse d' AVISPA, le résultat est montré par la Figure V.1 qui représente les traces d'attaques généré automatiquement en appuyons sur le bouton « Attaque simulation » (voir Annexe A) :

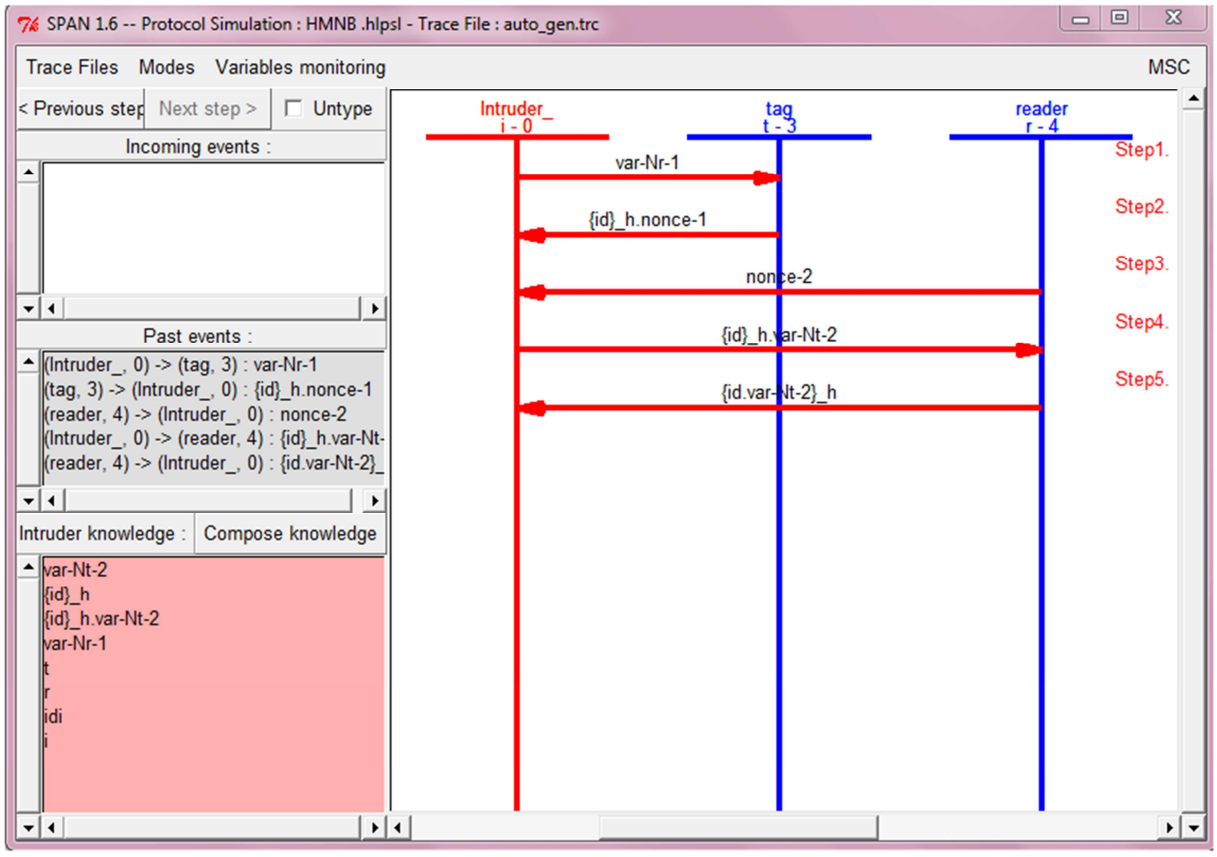


Figure V.1 : Trace d'attaque MITM de HMNB (CL-Atse)

Les traces d'attaques détectées par OFMC et CL-Atse montrent bien qu'il y a deux sessions de communication (lecteur-intru et intru-tag) d'où l'élimination de la communication avec le tag légitime. Ces attaques sont appelées attaque d'usurpation d'identité (attaque sur aut_tag) qui est similaire à celle découverte par les auteurs de [52]. Pour la confidentialité de l'ID et IDP et authentification du lecteur ont vois bien que les deux outils ne détectent aucune trace d'attaque, donc on peut déduire que ce protocole est sur pour ces deux buts.

Donc, d'après ces vérifications on conclut que HMNB n'est pas sûr. Il reste vulnérable aux attaques de rejeu et main in-the-middle. Cela, nous a motivé à proposer une nouvelle version pour HMNB qui sera plus sûr et plus résistante à ces attaques.

V.5 Proposition d'une amélioration du protocole HMNB

Comme nous l'avons vérifié formellement dans la section précédente, le protocole HMNB proposé dans [51] ne résiste pas aux attaques d'authentification (rejeu et main in-the-middle) contrairement à ce qu'ils ont affirmé les auteurs de [51]. Dans cette section, on va présenter un protocole d'authentification mutuelle qui est une amélioration du protocole HMNB. Pour laquelle on vérifie la confidentialité, l'authentification du tag et l'authentification du lecteur par les outils AVISPA et SPAN. Le protocole conçu protège la vie privée de l'utilisateur.

Pour évaluer ses performances, on le comparera avec d'autres protocoles RFID qu'on à déjà décrit dans le chapitre III [53] [54] [55].

V.5.1 Description du protocole

Selon l'ordre des messages transmis, le processus d'authentification mutuelle se déroule comme le montre la Figure V.2:

Etape 1 : Défi (Challenge) :

Le lecteur RFID génère un nombre aléatoire Nr et envoie ensuite une requête au tag. Trois cas peuvent se produire: 1) Aucun tag ne répond, 2) Un tag répond, 3) Beaucoup de tags répondent en même temps.

Etape 2 : Authentification de tag :

1. le tag recevant le Nr génère un nombre aléatoire Nt et vérifié son état si S=0 le tag calcule P := h(ID) sinon P := h(ID‖Nr‖Nt), et ce cas devrait se produire rarement dit impossible. Puis met à jour son état S :=1.

2. Le tag envoie P avec son nonce Nt comme dans le protocole HMNB en plus h(Nt||Nr) au lecteur.

3. Le lecteur renvoie P‖h(Nt || Nr) ‖Nt et son nonce Nr au serveur.

4. A partir de la base de données, le serveur compare P avec les valeurs de HID stockés dans la base de données ou avec h(ID || Nt || Nr) ou h(IDP || Nt || Nr) si les valeurs de HID ne correspondent pas. Si P est trouvé le serveur considère ID comme identité du T puis il vérifie H(Nt || Nr), Nr= Nr (celui envoyer par R et celui contenu dans H(Nt || Nr). Si ces égalités sont vérifiées, l'authentification du tag est effectuée avec succès, sinon le tag est rejeté.

Etape 3 : Authentification de lecteur:

1. Le serveur après la mise à jour d'ID, IDP et HID comme illustré dans le Tableau V.3, calcule et envoie au lecteur Q = H(IDP‖Nt) tel que IDP = IDi.

2. Le lecteur envoie le message Q comme dans HMNB accompagné de son nonce Nr au tag.

3. Le tag calcule Q'=H(IDP‖Nt) et vérifie si Q'=Q, Nt=Nt, Nr=Nr (celui envoyer par R). S'elles sont vérifiées l'authentification du lecteur est effectuée avec succès. le tag met à jour son ID := H (ID‖ Nr) puis remet son état à 0. Sinon l'authentification du lecteur est échouée.

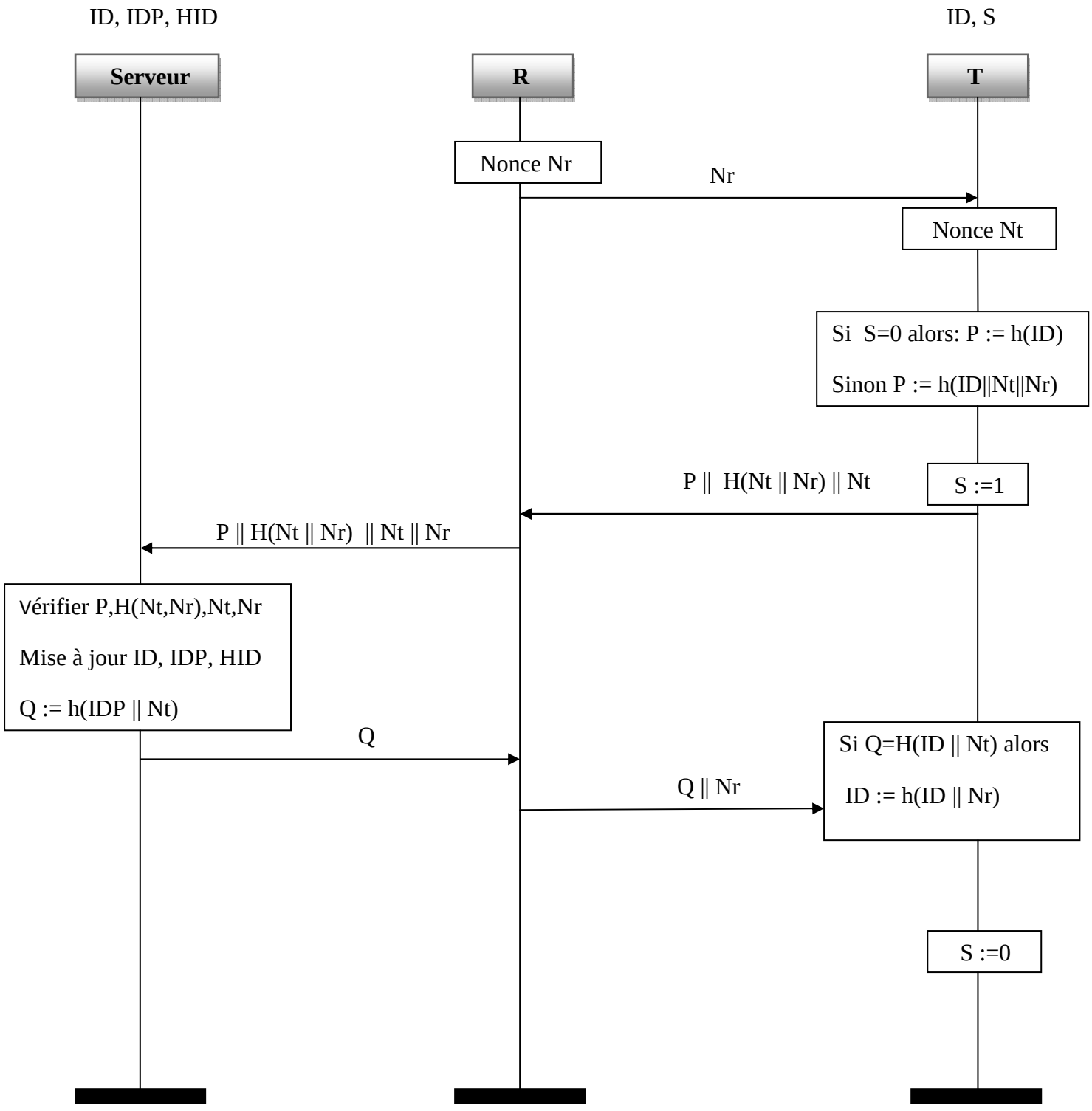


Figure V.2: Diagramme de séquences du protocole amélioré.

La réponse du tag	Les mises à jour (au niveau de la base de données)
h(ID), H(Nt.Nr), Nt	IDP := ID; ID := h(ID, Nr); HID := h(ID)
h(ID, Nt, Nr), H(Nt.Nr), Nt	IDP := ID; ID := h(ID, Nr); HID := h(ID)
h(IDP, Nt, Nr), H(Nt.Nr), Nt	ID := h(IDP, Nr); HID := h(ID)
Autres	Rejeter le tag

Tableau V.3: Mises à jour dans le protocole amélioré.

V.5.2 Analyse de ce protocole

V.5.2.1 Vérification automatique :

Deux attaques d'authentification possibles comme nous l'avons cité dans les sections précédentes :

- ***Attaque par rejeu***

La spécification suivante du rôle environnement représente une session entre un tag légitime et un lecteur légitime (t et r).

```
role environment() def=
const t,r : agent, id: text, h: hash_func,
aut_tag, aut_reader: protocol_id
intruder_knowledge = {t,r,h}
composition

%Attaque par rejeu
session(t,r,id,h)
end role
```

Le résultat de la vérification avec OFMC est le suivant :

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
```

```
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\HMNB amélioré.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 2 nodes
  depth: 1 plies
```

- ***Attaque main in-the-middle***

La spécification suivante du rôle environnement représente trois sessions entre un tag et un lecteur et un intrus (t, r et i).

```
role environment() def=
const t,r : agent, id,idi: text, h: hash_func,
aut_tag, aut_reader: protocol_id
intruder_knowledge = {t,r,idi,h}
composition
session(t,r,id,h)  /\ session(t,i,idi,h) /\ session(i,r,idi,h)
end role
```

Le résultat de la vérification avec CL-Atse est le suivant :

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  UNTYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\HMNB amélioré.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 4 states
```

Reachable : 0 states
Translation: 0.01 seconds
Computation: 0.00 seconds

❖ Ces résultat signifie en clair qu'il n y a pas d'attaques par rejeu et de main in-the-middle contrairement au protocole HMNB qui souffre d'attaque d'authentification de tag comme c'est montré dans la partie précédente.

V.5.2.2 Autre analyses :

A noter, il existe dans les systèmes RFID des propriétés de sécurité un peu spécifique qui ne peuvent pas être vérifiées avec des outils AVISPA & SPAN, comme la propriété de la désynchronisation.

- ***Résistance à la désynchronisation***

La désynchronisation est un type de dénis de service, notre protocole ne souffre pas de cette attaques contrairement à HMNB, voir Figure V.3, parce que le lecteur et le tag vérifie toujours l'intégrité des nonces Nr et Nt c'est la raison d'inclusion d'un terme supplémentaire H(Nt || Nr) dans le premier message envoyé par le tag et Nr dans le message Q, comme ça le lecteur et le tag vérifie toujours l'intégrité des valeurs de Nt et Nr, voir Figure V.4.

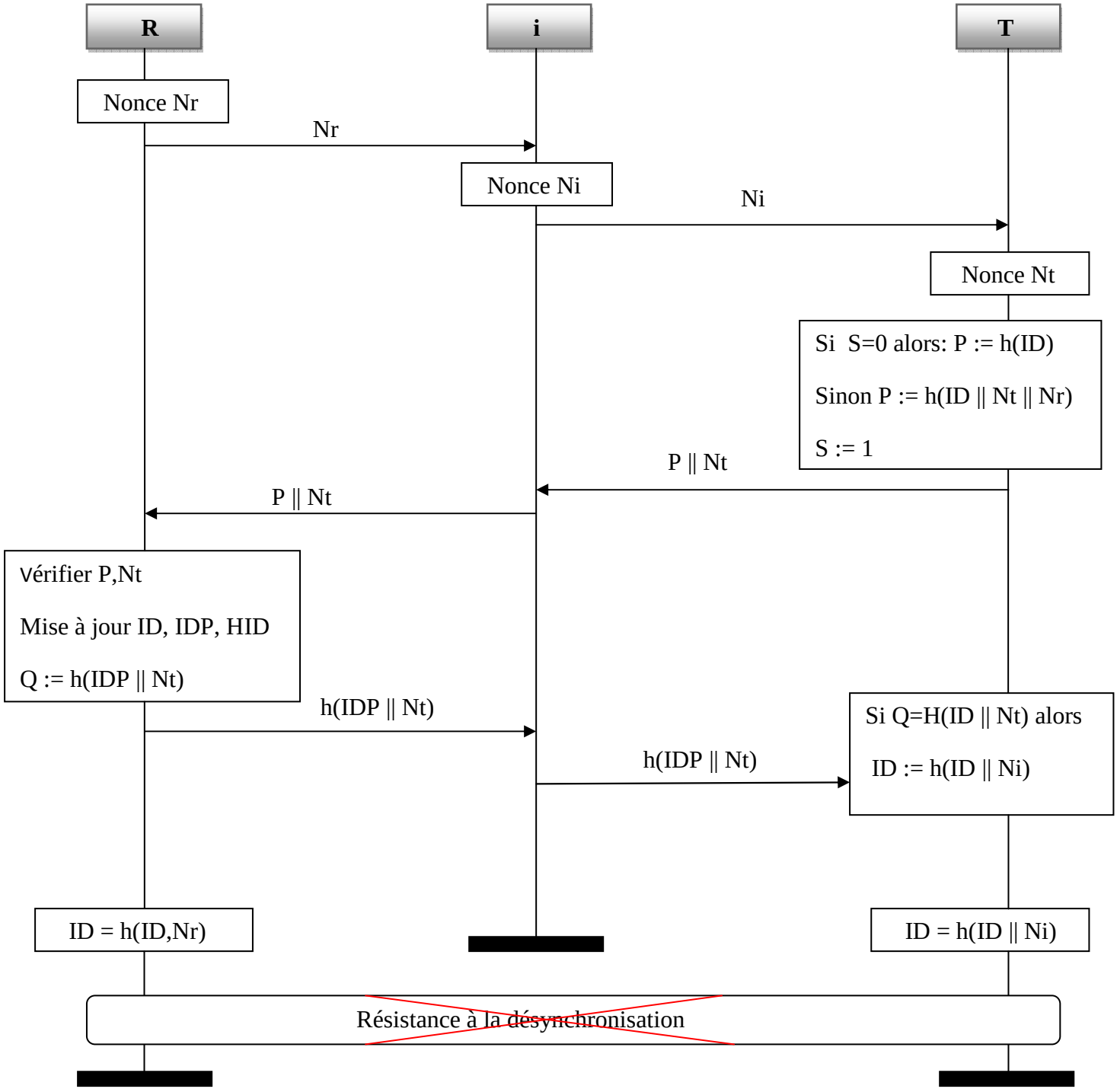


Figure V.3: Diagramme de séquences illustre l'attaque de désynchronisation dans HMNB.

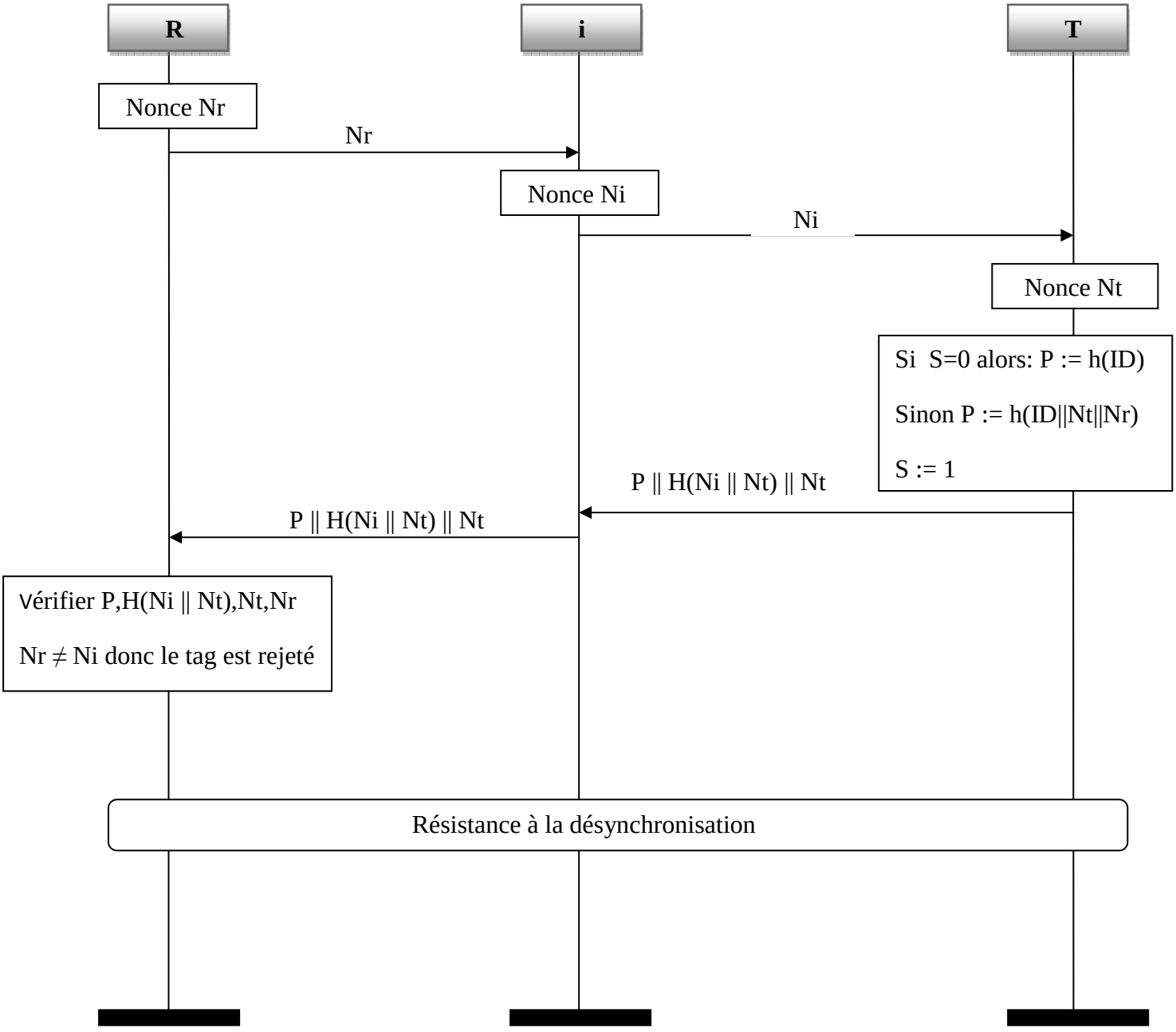


Figure V.4: Diagramme de séquences illustre la résistance à l'attaque de désynchronisation dans le protocole amélioré.

V.5.3 Comparaison de la sécurité

Le Tableau V.4 illustre une comparaison de la sécurité avec les protocoles décrit dans le chapitre III:

Protocoles / Attaques	CH [53]	LAK[54]	HMNB [51]	LMAP [55]	Notre protocole
Résistance à l'attaque par rejeu	non	non	non	non	oui
Résistance à MITM	oui	oui	non	non	oui
Résistance à la désynchronisation	oui	oui	non	non	oui

Tableau V.4 : Analyse de la sécurité

V.5.4 Analyse de la performance

Comme nous l'avons vu dans le Chapitre I, le coût et la limitation des ressources dans les systèmes RFID représentent les défis à prendre en charge lors de la réalisation d'un tel système, le tableau V.5 illustre le coût de calcul, l'espace de stockage et le coût de la communication entre le tag et le lecteur et la complexité de calcul au niveau de la base de données. Le coût de calcul est en fonction du nombre d'opérations de la fonction de hachage dans la phase d'authentification mutuelle.

Performance \ Protocole		CH[53]	LAK[54]	HMNB[51]	LMAP[55]	Notre protocole
Coût de calcule	Tag	1h	2h	2h	--	3h
	Lecteur	1h	2h	1h	--	1h
PRNG	Tag	1g	1g	1g	--	1g
	Lecteur	1g	1g	1g	2g	1g
Espace de Stockage	Tag	2l	1l	2l	6l	2l
	Lecteur	2l	2l	3l	6l	3l
Coût de la communication	R -> T	2½l	2l	2l	4l	3l
	T -> R	1½l	2l	2l	2l	3l
	∑	4l	4l	4l	6l	6l
Klé symétrique		1l	1l	--	4l	--
Complexité de calcule dans la base de données		O(1)	O(1)	O(1)	O(1)	O(1)

Notations : h : nombre d'opération de la fonction de hachage ;
g : nombre d'opération du générateur de nombre aléatoire ;
l : taille par bit pour chaque variable ;
-- : La valeur n'existe pas.

Tableau V.5 : Analyse des performances.

Après la comparaison de la sécurité et les performances de protocole proposé par rapport à d'autres protocoles, on peut conclure que notre protocole est sûr et adapté aux systèmes RFID.

V.6 Conclusion :

Dans ce chapitre, nous avons vérifié formellement les deux protocoles cryptographiques R^2AP et HMNB 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 R2AP est sûr, par contre le HMNB est vulnérable vis-à-vis des attaques par rejeu et main in-the-middle. Pour cela, nous avons proposé une version améliorée de HMNB qui résiste à ces deux types d'attaques, mais cela ne permet en aucun cas de confirmer que ces protocoles sont totalement valides à cause de la présence de quelques propriétés particulières difficiles à vérifier d'une manière automatique.

Conclusion générale

Nous nous sommes intéressés, dans le cadre de ce mémoire, à la vérification formelle des protocoles cryptographiques dans les systèmes RFID. Ces derniers constituent un domaine de recherche très riche et peuvent être appliqués dans plusieurs domaines différents « militaire, médical, agriculture, etc. ». Néanmoins, il reste encore de nombreux défis à résoudre dans ce domaine afin de pouvoir les utiliser dans des conditions plus sécurisés. L'un des problèmes qu'on peut rencontrer dans ce genre de réseaux est la problématique de la sécurité, elle est causée par le fait que les RFIDs emploient des protocoles de sécurité vulnérables. Pour cela, le réseau sera exposé à plusieurs types d'attaques internes ou externes. Ce mémoire a pour objectif de bien comprendre la vérification formelle des protocoles RFID à l'aide d'outils automatique AVISPA et SPAN, qui sont utilisés dans divers domaines grâce à leurs efficacités.

Dans notre travail, nous avons présenté quatre chapitres. Le premier chapitre a été dédié à une présentation générale des systèmes RFID. Les différentes technologies RFID existantes ont été introduites et les défis de la technologie RFID ont été mis en évidence.

Dans le deuxième chapitre, les attaques et la sécurité dans les systèmes RFID ont été exposées. Nous avons présenté chaque attaque et ses contre mesures mis en œuvre et nous avons abordé le problème de la sécurité de la vie privée qui représente les pierres cruciale des systèmes RFID.

Le troisième chapitre a concerné la vérification formelle et ses outils. Ce chapitre a été illustré par la présentation d'outil AVISPA : HLPSL et SPAN ont été introduites.

Finalement, dans le quatrième chapitre nous avons présenté la vérification de deux protocoles RFID proposé par [50], [51] respectivement, et nous avons proposé une amélioration pour le protocole HMNB pour résoudre les problèmes détectés.

Ce projet nous a été d'un grand apport pédagogique puisqu'il nous a permis d'approfondir nos connaissances dans le domaine des systèmes RFID principalement la sécurité et de nous familiariser avec l'environnement de travail SPAN et HLPSL que nous avons découvrent pour la première fois.

Annexe : L'interface graphique SPAN pour AVISPA.

Annexe A : L'interface graphique SPAN pour AVISPA.

A.1 Environnement de travail

A.1.1 Système d'exploitation

Nous avons développé notre projet, sous le système d'exploitation Windows, avec le langage HLPSL, il peut être intégré dans n'importe quel autre système d'exploitation supportant le SPAN.

A.1.2 Présentation et installation de SPAN

SPAN est développé en Ocaml et Tcl / Tk version 8.4 (pour Linux et MacOS) et version 8.3 (pour Windows) est nécessaire pour utiliser cette application. Il est distribué sous licence LGPL et est disponible gratuitement sur http://www.irisa.fr/lande/genet/span/.

SPAN est un animateur de protocoles, son rôle est d'aider à concevoir des caractéristiques formelles. Il peut animer des spécifications HLPSL, à savoir produire un message de séquence (Message Sequence Charts pour MSC) de manière interactive qui peut être considérée comme une notation "Alice & Bob" d'une spécification HLPSL et peut représenter un ou plusieurs sessions du protocole en parallèle selon les informations données dans le rôle environnement. Cet outil utilise certaines bibliothèques d'AVISPA comme illustre la Figure A.1: une pour HLPSL traduction, conçu par Laurent Vigneron, et une pour la théorie de filtrage modulo xor et exponentielle, conçues par Mathieu Turuani. SPAN est le premier outil conçu pour aider à écrire, animer, et comprendre des spécifications HLPSL [76].

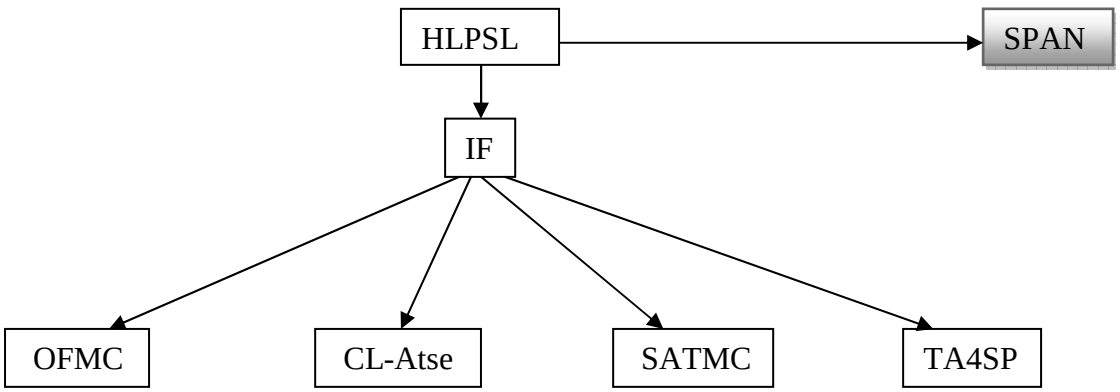


Figure A.1 : Architecture du système AVISPA et SPAN.

A.1.3 Le langage de programmation HLPSL

Ce langage modélise un protocole sous la forme d'un graphe d'états. Il permet de spécifier des rôles, les différents messages échanges ainsi que les propriétés de sécurité à respecter, il permet également de déclarer la connaissance de l'intrus dans l'environnement.

A.1.4 Interface graphique pour AVISPA

SPAN est livré avec une version locale de l'interface graphique Web d'AVISPA comme la montre la Figure A.2. Il fournit les mêmes caractéristiques: édition simple des spécifications du protocole, la sélection et la configuration de la vérification, les back-ends d'AVISPA et trois boutons : Protocol simulation (les agents Alice et Bob seulement), Intruder simulation (les agents Alice, Bob et un intrus) et Attack simulation (même disposition que la Intruder simulation mais les attaques sont automatiquement construits en utilisant OFMC ou CL-ATSE) [76] [75].

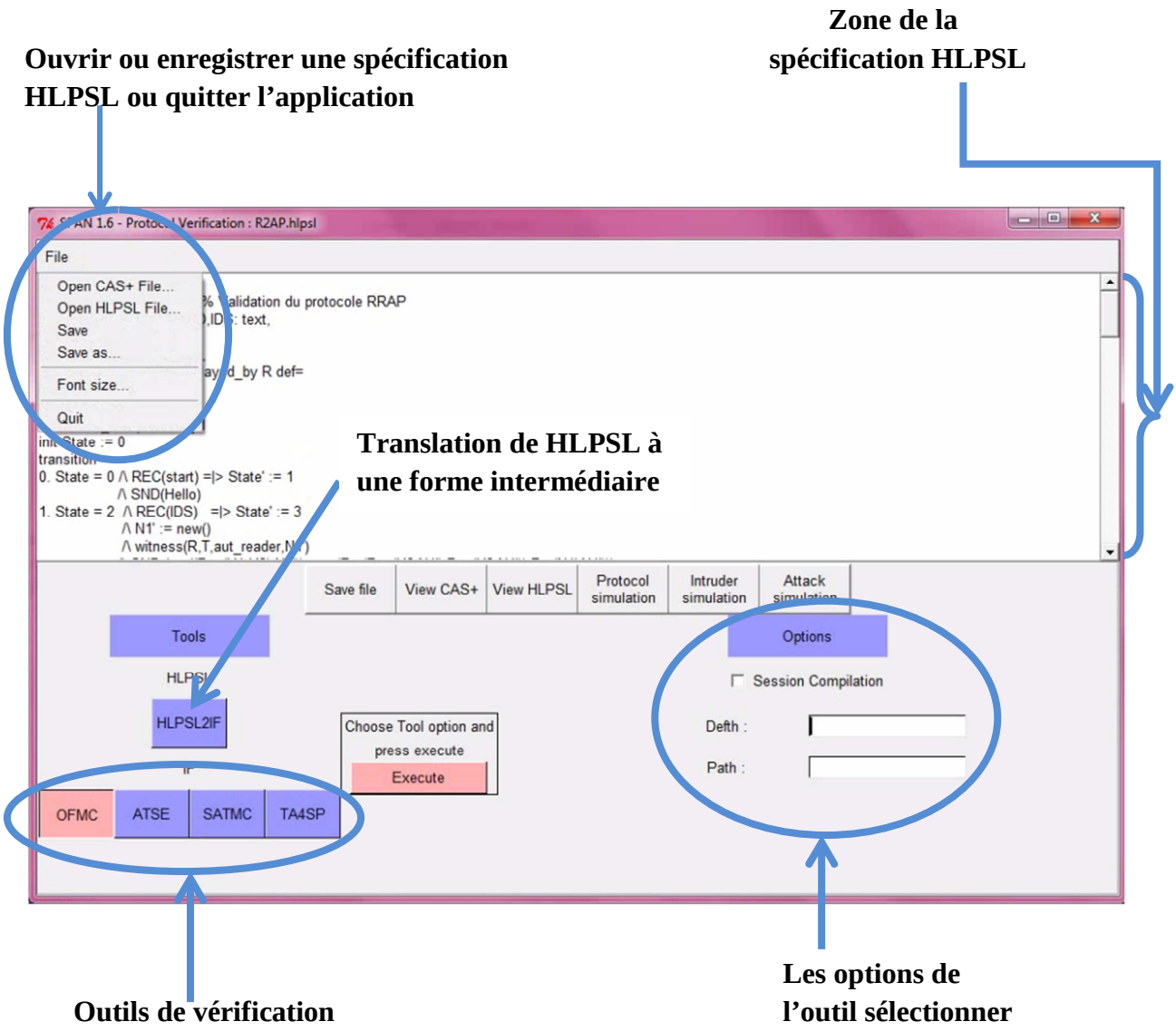


Figure A.2 : Interface graphique de vérification.

Annexe A : L’interface graphique SPAN pour AVISPA.

A partir d’une spécification HLPSL, SPAN construit un MSC correspondant à cette spécification. SPAN peut représenter une ou plusieurs sessions de protocole en parallèle selon les informations données dans le rôle environnement. Ensuite, les MSCs sont réalisés de manière interactive avec l'utilisateur. A chaque instant, SPAN propose à l'utilisateur de choisir entre toutes les transitions pour lesquelles un message peut être envoyé par un agent et reçu par un autre.
Cette approche permet de résoudre de manière interactive tous les choix qui peuvent survenir au cours de la construction du MSC. L'exécution de la transition d'un protocole ajoute généralement une transition sur la MSC.
Les Figures A.3 et A.4 montre une exécution partielle d'un protocole où les événements entrants contiennent tous les transitions qui peuvent être actuellement choisi (par double clic), et les événements passés contiennent la liste de toutes les transitions déjà choisi.

SPAN comprend également la possibilité de vérifier les valeurs des variables de chaque agent à tout instant : l'utilisateur peut choisir les variables de chaque rôle qu'il veut suivre. Il est possible de contrôler les valeurs des variables dans le menu "Variables monitoring".

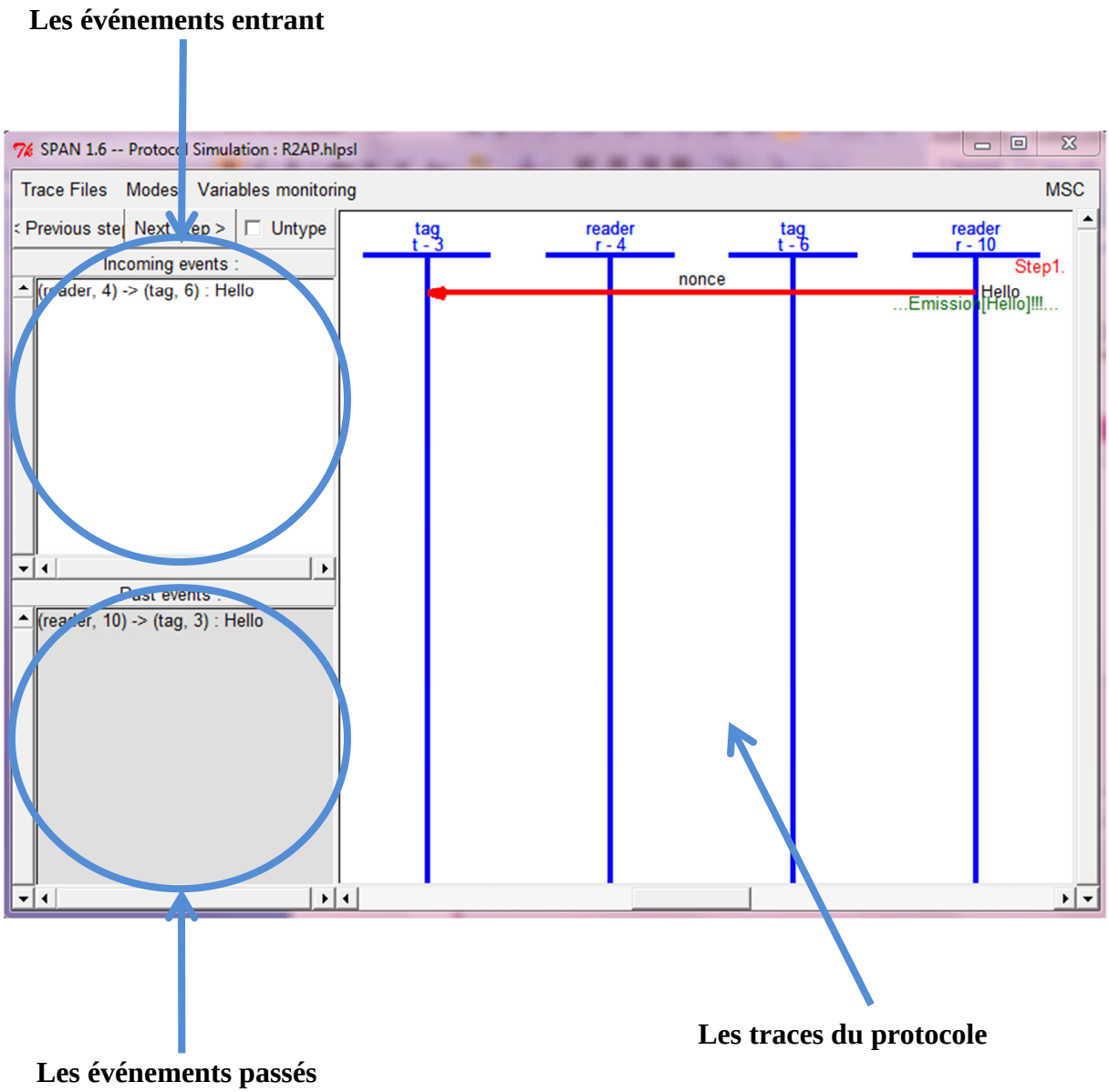


Figure A.3 : L'interface SPAN lors d'une exécution interactive (Protocol simulation).

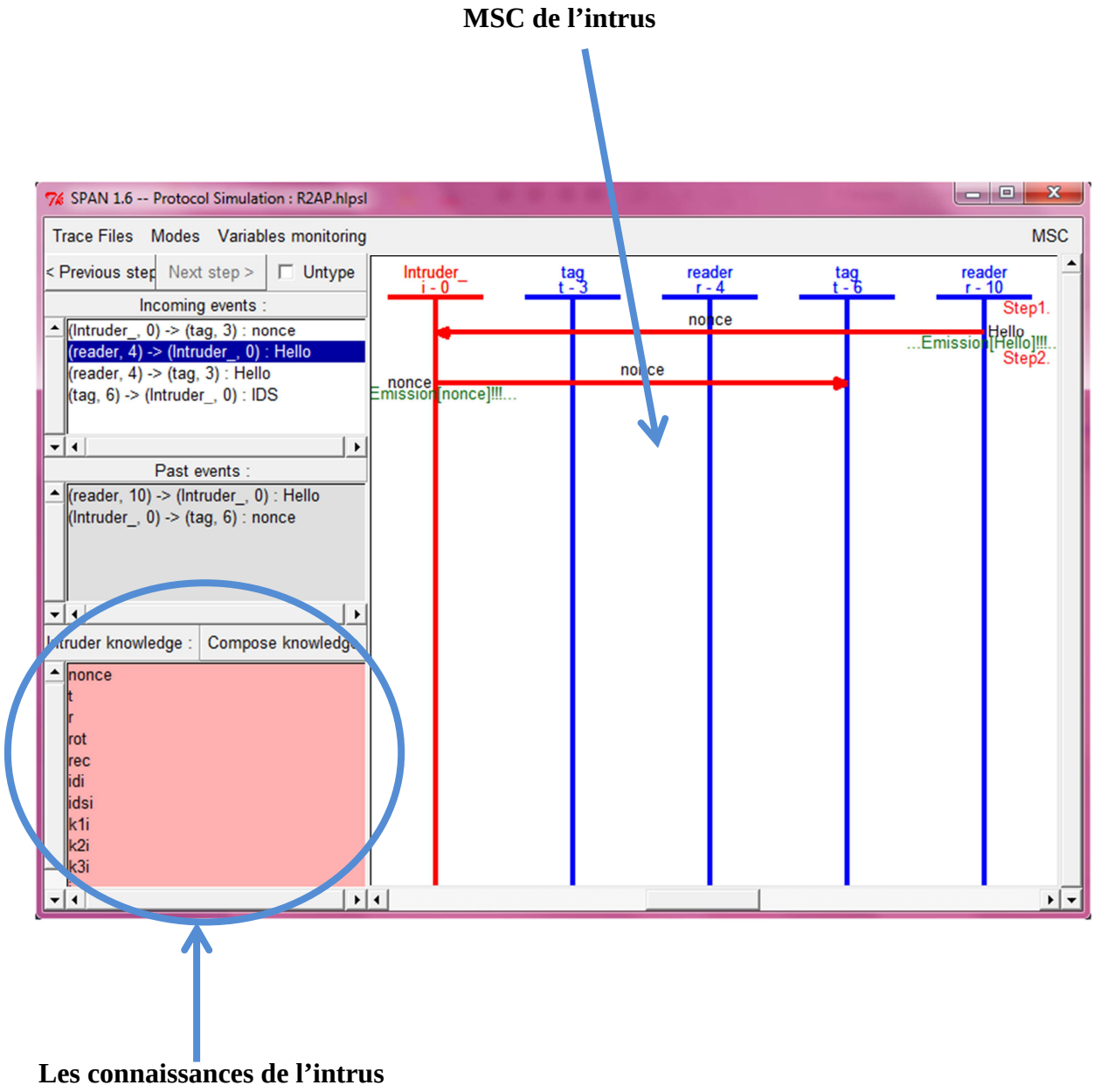
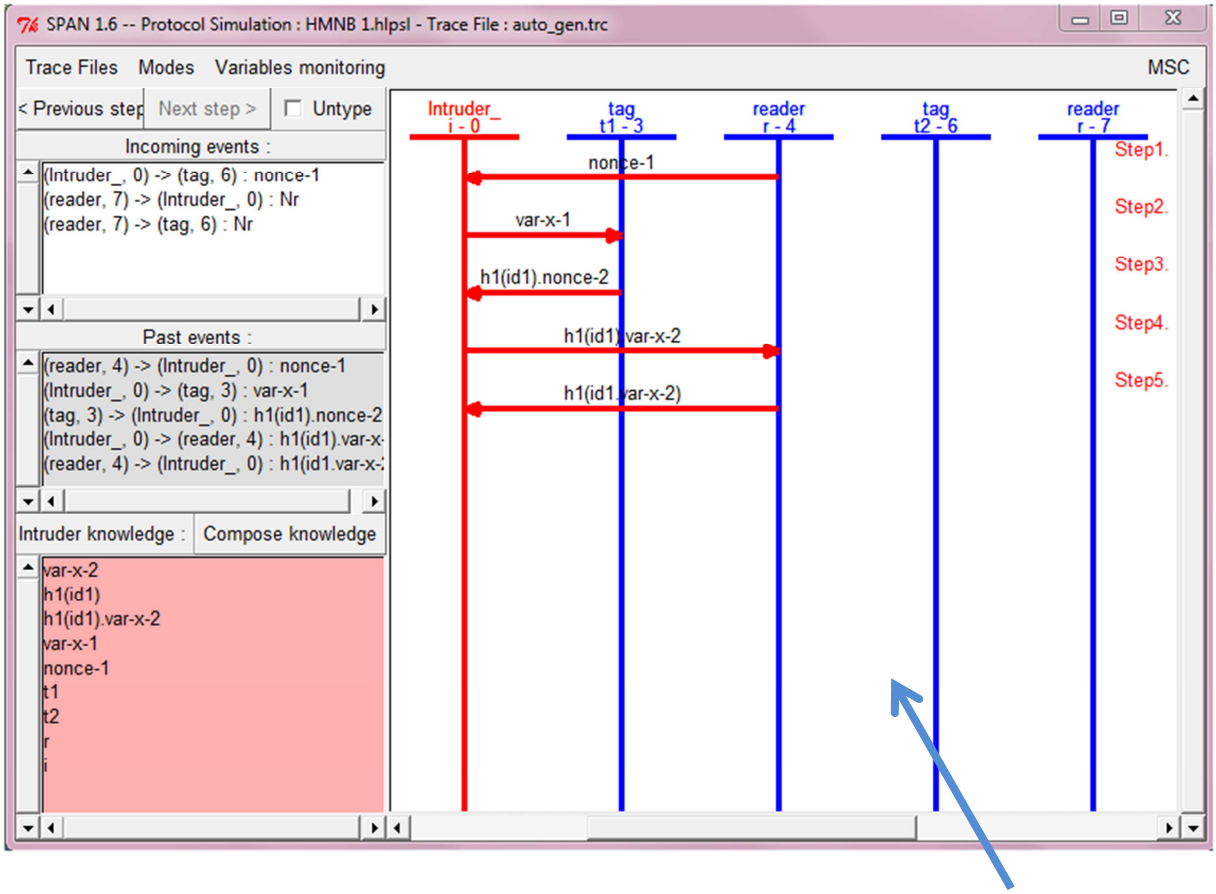


Figure A.4 : L'interface SPAN lors d'une exécution interactive (Intruder simulation).

La Figure A.5, décrit l'interface de simulation d'intrusion (Attack simulation) générée automatiquement lors de la validation du protocole HMNB. Le but de cette interface est de fournir un moyen de construire des attaques sur les protocoles. En effet, les outils de vérification automatique finissent toujours avec la plus petite attaque qui peut être trouvé sur un protocole. Notez que les attaques trouvées par OFMC ou CL- ATSE peut être automatiquement chargé en tant que MSC dans l'interface Attack simulation.



La trace d'attaque générée lors de la vérification de HMNB avec OFMC.

Figure A.5 : Présentation de l'interface Attack simulation.

A.2 Présentation de quelques codes sources

Dans cette section on détaille les rôles basiques et composés de la spécification du R^2AP en langage de spécification HLPSL.

a. On présente maintenant les rôles basiques: déclarations des paramètres et les variables locales:

```
role reader ( R,T: agent, ID,IDS: text,
Rot,Rec: hash_func,
K1,K2,K3:symmetric_key,
SND,REC: channel(dy)) played_by R def=
local State : nat,
N1,N2,Hello : text
const sec_ids: protocol_id
init State := 0
transition
0. State = 0 /\ REC(start) =|> State' := 1
                /\ SND(Hello)
1. State = 1  /\ REC(IDS)   =|> State' := 2
                /\ N1' := new()
                /\ witness(R,T,aut_reader,N1')
               /\ SND (xor((Rec(K1.K2).N1')) .
xor(Rot(Rec(K2.N1').Rec(K3.N1')).Rot(N1'.
N1')))
2. State = 2 /\ REC
(xor(Rec(Rec(K2.K3).Rec(N1.N1)).ID))
 =|> State' := 3  /\ N2' := new()
/\ witness(R,T,aut_reader,N2')
/\SND(xor(Rec(N1'.K3).Rec(K1.K3).N2').
xor(Rot(Rec(K2.N2').Rec(K2.N1')).Rot(N2'.
N2')))
   /\ IDS':= xor(Rec(xor(IDS.N2).K3).K1)
   /\ K1':= xor(Rec(N2.N1).K2)
   /\ K2':= xor(Rec(K2.xor(N1.N2)).K3)
   /\ K3':= xor(Rec(K2.K3).N1)
   /\ secret(IDS',sec_ids,{R,T})
   /\ request(R,T,aut_tag,ID)
end role
```

```
role tag ( T,R: agent,
ID ,IDS: text,
Rot,Rec : hash_func,
K1,K2,K3:symmetric_key,
SND,RCV: channel(dy)) played_by T def=
local State : nat,
N1, N2 ,Hello: text
const sec_id: protocol_id
init State := 0
transition
1. State = 0 /\ RCV(Hello) =|>
State' := 1 /\ SND (IDS)
2. State = 1 /\ RCV(xor((Rec(K1.K2).N1')) .
xor(Rot(Rec(K2.N1').Rec(K3.N1')).Rot(N1'.
N1'))) =|>State' := 2
/\SND(xor(Rec(Rec(K2.K3).Rec(N1.N1)).ID
))
   /\ secret(ID,sec_id,{T,R})
   /\ request(T,R,aut_reader,N1)
  /\ witness(T,R,aut_tag,ID)
3.State = 2
/\     RCV(xor(Rec(N1'.K3).Rec(K1.K3).N2').
xor(Rot(Rec(K2.N2').Rec(K2.N1')).Rot(N2'.
N2')))   =|>  State' := 3
/\  IDS':= xor(Rec(xor(IDS.N2).K3).K1)
/\ K1':= xor(Rec(N2.N1).K2)

/\K2':=xor(Rec(K2.xor(N1.N2)).K3)
 /\ K3':= xor(Rec(K2.K3).N1)
 /\ request(T,R,aut_reader,N1,N2)


end role
```

Ce code représente la spécification du rôle lecteur et tag, dont on a :

- R, T : des agents jouant les rôles reader et tag respectivement.
- SND, REC : deux canaux pour l'émission et la réception des messages respectant le modèle Dolev & Yao.
- sec_id, sec_ids : sont des constantes permettant d'identifier la propriété à vérifier, ici la confidentialité des variables ID et IDS.
- IDS : contient l'index du tableau ou l'identificateur du tag est stocké.
- State := 0 : la variable State initialement instanciée par la valeur 0.
- Agent, text, hash_fuc, channel (dy), nat, symmetric_key, protocol_id: sont des types.

Le premier caractère des noms des variables est majuscule, et pour les constantes, les types et les mots-clé, le premier caractère est minuscule en HLPSL. La notion de présence du "prime" dans la fin d'une variable est définie par : si un agent souhaite d'affecter une nouvelle valeur à X à partir du contenu d'un message entrant, on notera X'. Dans le cas inverse, si un agent cherche à comparer sa valeur courante de X avec une valeur émise dans le message, on notera X.

b. On passe à la représentation des rôles composés:

```
role session(T,R : agent,ID,IDS : text,Rot,Rec
: hash_func, K1,K2,K3:symmetric_key) def=
local St,Rt,Sr,Rr : channel(dy)

composition
tag(T,R,ID,IDS,Rot,Rec,K1,K2,K3,St,Rt) /\
reader(R,T,ID,IDS,Rot,Rec,K1,K2,K3,St,Rt)
end role
```

```
role environment() def=
const t,r : agent,
id,ids: text,
rot,rec:hash_func,
k1,k2,k3:symmetric_ey,
aut_tag, aut_reader:protocol_id
intruder_knowledge = {t,r,rot,rec}

composition

% Attaque par rejeu

session(t,r,id,ids,rot,rec,k1,k2,k3)

end role
```

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

Le rôle environement s'intéresse au scénario de vérification suivant : deux sessions du protocole en parallèle noter par le symbole /\ concernant un tag légitime et un même lecteur légitime. Les connaissances initiales de l'intrus sont les noms d'agents t qui représente le tag, et l'agent r qui représente le lecteur. Cette spécification permet de détecter les attaques attaque par relais s'il existe. Les données aut_reader et aut_tag sont des constantes qui permettent d'identifier les propriétés d'authentification du lecteur et d'authentification du tag respectivement.
La section goal nous permet de spécifier les objectifs de sécurité permettant aux outils AVISPA de faire une recherche sur les attaques.

Références bibliographique :

[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.

[4]: Klaus Finkenzeller, John Wiley and Sons. << The RFID Handbook >>.Book, Munich, Germany, second edition, DEC 2003.

[5]: OCDE. << RFID identification par radio fréquence >>. Réunion ministérielle de l'OCDE le futur de l'économie internet, Séoul, Corée, 17-18 juin 2008.

[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.

[9]: Noyel Mélanie et Pisaneschi Thomas. << Mise en place d'un système RFID pour une entreprise de panneaux laqués haute finition >>. Rapport de fin d'étude, université de nancy, 2010 / 2011.

[10]: http://www.jesic-tech.com/RFID_tag.html

[11]: Qinghan XIAO, Thomas GIBBONS et Hervé LEBRUN. << RFID Technology, Security Vulnerabilities, and Countermeasures>>, Source: Supply Chain, The Way to Flat Organisation, Book edited by: Yanfang Huo and Fu Jia, ISBN 978-953-7619-35-0, pp. 404, I-Tech, Vienna, Austria, December 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 Computer and Electrical Engineering, Vol.3, No.1, February, 2011

[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/PAC-ID/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.

[28]: Myrto Arapinis. << Sécurité des protocoles cryptographiques : décidabilité et résultats de réduction >>. Thèse de doctorat, université Paris-Est, 20 Novembre 2008.

[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.

[42]**:** Gerhard Hancke. <<A practical relay attack on ISO 14443 proximity cards>>, university of Cambridge, Computer Laboratory JJ Thomson Avenue, Cambridge CB3 0FD, UK, 1 Project report, January 2005.

[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.

[60]: M. Rafik KHEDDAM. << Génération de tests de protocoles cryptographiques par mutations de modèles >>. Mémoire de fin d'étude en vue de l'obtention du grade de Master2, université de FRANCHE-COMITE. 18 juin 2010.

[61]: Stephen H. Brackin, Dromquinna Manor, Kenmare, County Kerry << A HOL extension of GNY for automatically analyzing cryptographic protocols >>. In Ninth IEEE Computer Security Foundations Workshop, March 10 - 12, 1996, Ireland, page 62.

[62]: Kieran Healy, Tom Coffey, and Reiner Dojen. << A comparative analysis of state-space tools for security protocol verification >>. In WSEAS Transactions on Information Science and Applications, Volume 1, Issue 5, pages 1256–1261, 2004.

[63]: D. Dolev and A. C. Yao. <<On the security of public key protocols>>. In Proceedings of the 22Nd Annual Symposium on Foundations of Computer Science, SFCS, Computer Society IEEE Washington, USA, 1981. pages 350–357

[64]: European Community under the Information Society Technologies Programme. <<HLPSL Tutorial, A Beginner's Guide to Modelling and Analysing Internet Security Protocols The AVISPA team >>. IST-2001-39252, Document Version: 1.1, June 30, 2006. www.avispa-project.org

[65]: Bruno Blanchet, Martín Abadi, and Cédric Fournet.<< Automated verification of selected equivalences for security protocols >>.Journal of Logic and Algebraic Programming, 2007

[66]: L. Bozga, Y. Lakhnech, and M. Périn. <<Pattern-based abstraction for verifying secrecy in protocols >>.International Journal on Software Tools for Technology Transfer (STTT),8:57-76, Feb 2006.

[67]: Lowe. << Casper : A Compiler for the Analysis of Security Protocols. In PCSFW : Proceedings of The 10th Computer Security Foundations Workshop >>. IEEE Computer Society.journal, 1997.

[68]: Aribi Noureddine. <<Vérification des Protocoles cryptographiques avec la Programmation>>. Thèse de doctorat, université d'Oran Faculté Des Sciences, janvier 2008.

[69]: Alix Trieu, Amélie Royer , Antoine Chatalic, Baptiste Tessiau, Lucas Seguinot, Siargey Kachanovich . <<Aide à la conception et vérification de spécifications formelles de protocoles cryptographiques >>. Université de Rennes. prénom.nom@ens-rennes.fr

[70]: European Community under the Information Society Technologies Programme. <<Automated Validation of Internet Security Protocols and Applications>>. User Manual, IST-2001-39252, Document Version: 1.1, June 30, 2006. www.avispa-project.org

[71]: Yann GLOUCHE, Thomas GENET and Erwan HOUSSAY. << a Security Protocol ANimator for AVISPA>>. User Manual, IRISA/ Universit_e de Rennes, Document Version: 1.5, September 2008. www.avispa-project.org