Vérification automatique des protocoles d'authentification des systèmes RFID

Noureddine Chikouche
Ecole Doctorale STIC
Université de M'sila
Algérie
chiknour28@yahoo.fr

Mohamed Benmohammed
Département d'informatique
Université de Constantine
Algérie
ben_moh123@yahoo.com

Résumé — **Le canal de communication entre le tag et le lecteur dans la technologie d'identification par radiofréquence (RFID) est insécurisé, ce qui le rend ouvert devant les attaques logiques sur le protocole de sécurité. Dans ce papier on va étudier des protocoles d'authentification dans les systèmes RFID qui peuvent être modélisés avec le langage de spécification HLPSL. Le travail se focalise sur la vérification automatique des propriétés de sécurité, la confidentialité et de l'authentification avec les outils AVISPA, ainsi que sur la comparaison de la complexité d'implémentation des primitives cryptographiques exigées dans ces protocoles sur les tags.**

Mots clés : AVISPA; Protocole d'authentification; RFID ; Confidentielité ; Primitive cryptographique.

I. INTRODUCTION

Parmi les systèmes qui ont été développés rapidement au cours des dernières années, on peut constater ceux d'identification par radiofréquence (RFID) qui sont utilisés dans des domaines divers. La caractéristique principale d'un système RFID est que son utilisation des ressources informatiques est limitée d'une part, tels que : la mémoire, le processeur, la consommation d'énergie, etc. Et d'une autre part, les systèmes RFID sont nécessaires pour assurer la sécurité dans toutes les couches : la couche physique, la couche réseau-transport (où se trouvent les protocoles de sécurité), et la couche d'application.

La vérification de la sécurité des protocoles cryptographiques dépend généralement de deux axes complémentaires : la recherche d'une attaque et la preuve d'un protocole sûr. Dans le domaine de vérification automatique des protocoles de sécurité, Il y a plusieurs analyseurs de protocoles, mais la plateforme AVISPA (Automated Validation of Internet Security Protocols and Applications) [1] est l'analyseur le plus connu qui modélise un grand nombre de protocoles (79 protocoles). L'efficacité d'Avispa a été testée sur de nombreux protocoles récemment standardisés, par exemple par l'IETF (Internet Engineering Task Force) et des protocoles du domaine e-business.

Notre travail s'articule sur la vérification des protocoles d'authentification des systèmes RFID en utilisant des outils AVISPA après avoir modélisé ces protocoles en HLPSL. Une tâche complémentaire consiste à comparer la complexité d'implémentation des primitives cryptographiques exigées dans ces protocoles sur les tags.

Notre papier est divisé en plusieurs sections: la section II consiste à présenter les notations agréés pour décrire un protocole, La section III consiste à définir un système RFID ainsi que ses composants et ses applications. La section IV présente la plateforme AVISPA. Dans la section V on étudie des protocoles d'authentification mutuelle qui figurent sur les systèmes RFID par vérification des propriétés de sécurité. La section VI comporte une analyse des résultats obtenus dans la section précédente. La section VII consiste à décrire et à comparer les protocoles de côté de la complexité du tag. Et enfin, le papier est terminé par une conclusion générale.

II. NOTATIONS

Les notations de type Alice-Bob (voir Table. I) qui permet de décrire informellement beaucoup de protocoles cryptographiques.

TABLE I. NOTATIONS

Symbole	Signification
R , T	Nom d'agent honnête (un partcipant honnête du protocole), R : Lecteur ; T : Tag
m	Message
Nt, Nr	Nonce (nombre aléatoire « frais »)
h	fonction de hachage
,	Concaténation
K	Clé symétrique partagée entre R et T
ID	Identificateur partagé entre le tag et le lecteur
{m}_K	Le message m crypté avec K
R → T : m	R envoi un message m à T

III. LES SYSTMES RFID

L´identification par radiofréquence (RFID) est une technologie sans contact, cette technologie permet d'identifier un objet, d'en suivre le cheminement et d'en connaître les caractéristiques à distance grâce à une étiquette émettant des ondes radio. Le système RFID se compose de: *(1)* le tag (l'étiquette, transpondeur), *(2)* le lecteur *(3)* et le serveur (base de données, back-end) [2].

- Le tag se constitue d'une puce qui stocke les données et une antenne qui assure la communication entre le tag et le lecteur par radiofréquence. La plupart des tags

sont identifiés par un identificateur unique nommé EPC (Electronic Product Code). Le tag peut être actif ou passif. Un tag actif comporte une source d'alimentation (comme une pile) et émet un signal RF. Un tag passif capte son alimentation à partir du signal d'interrogation d'un lecteur.

- Le lecteur est un appareil qui communique sans fil avec des tags pour identifier l'élément connecté.
- Le serveur est un sous-système de traitement de données qui utilise les données obtenues à partir du lecteur à des fins utiles.

Le mécanisme de travail de ce système est défini comme suit : le lecteur RFID envoie un signal à radio sur une fréquence déterminée, le tag qui se trouve dans le champ d'action du lecteur utilise ce signal sous forme d'énergie, cette dernière alimente la puce ce qui permet de renvoyer les informations qu'elle contienne.

Les applications de RFID peuvent être utilisées dans les entreprises, par les individus ainsi que par les états. Ce système est exploité dans plusieurs domaines : le transport, la sécurité, la santé, la bibliothèque, et la logistique qui sont autant des domaines dans lesquels cette technologie existe déjà et apparaîtra dans le futur.

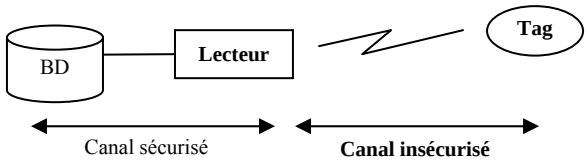

Figure 1. Le système RFID

La communication entre le serveur et le lecteur est sécurisée, c'est-à-dire que le canal de communication est un canal privé. Contrairement à ceci, la communication entre le tag et le lecteur est quant à elle insécurisée, c'est-à-dire que le canal de communication est un canal public (voir figure 1). Donc l'attaque à ce niveau est possible par l'écoute, car un signal radio peut être capté relativement et facilement par des personnes.

IV. LA PLATEFORME AVISPA

En juillet 2005 les partenaires du projet européen AVISPA ont publié leurs travaux de développement d'une plateforme contenant quatre outils d'analyse de protocoles et permettant la détection des attaques logiques sur les protocoles de sécurité. Cette plateforme suggère aussi des améliorations assurant la validité des propriétés de confidentialité et d'authentification. Les techniques de vérifications utilisées par AVISPA sont des techniques fondées sur le principe du Model-checking.

Les quatre outils AVISPA sont: OFMC (On-the-fly Model-Checker), CL-ATSE (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) et TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols).

A. HLPSL

HLPSL (High Level Protocol Specification Language) [3] est un langage formel de spécification modulaire, expressif et est basé sur des descriptions de rôles. Il supporte des primitives cryptographiques différentes (Clés symétriques et asymétriques, les fonctions de hachage) et de leurs propriétés algébriques (ou exclusif, exposant). Le but de ces spécifications étant de pouvoir vérifier des propriétés de sécurités, l'authentification et la confidentialité.

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

Les spécifications HLPSL de protocoles sont divisées en rôles, ces derniers sont répartis en deux catégories distinctes : les rôles dits basiques et les rôles dits de composition. Le premier type représente les agents participants aux protocoles, tandis que le second représente les scénarios des rôles basiques. A la fin de la spécification, on détermine les propriétés de sécurité à vérifier.

B. Les hypothèses de vérification :

Dans le cadre de la modélisation de protocoles de sécurité, il est nécessaire de modéliser également l'intrus, c'est-à-dire de définir son comportement et de le limiter. Pour cela, les hypothèses utilisées sont rassemblées sous le nom de « modèle de Delev-Yao » [5]. Ce modèle est basé sur deux hypothèses importantes qui sont: *le chiffrement parfait* et *l'intrus est le réseau*.

Le chiffrement parfait assure en particulier qu'un intrus ne peut déchiffrer un message `m` chiffré avec une clé `k` que s'il possède l'inverse de cette clé. La seconde hypothèse "l'intrus est le réseau" signifie que l'intrus peut intercepter et remplacer les messages envoyés par les acteurs honnêtes du protocole, et leur envoyer des messages sous une fausse identité.

Les transmissions des messages dans les canaux de communications sont publics ou privés. Dans les canaux publics les messages échangés sont connus par tout le monde, qu'ils soient honnêtes ou pas. Mais contrairement, les canaux privés sont des canaux définis entre certains participants honnêtes. Par conséquent, un intrus ne peut pas donc écouter les messages qui circulent sur ce genre de canaux.

Pour l'hypothèse "l'intrus est le réseau", le réseau du système RFID dans ce cas est sans fil. Il est basé sur la communication par des ondes radiofréquences.

C. Les propriétés à vérifier

On doit vérifier les propriétés de sécurité suivantes :

- La confidentialité : on peut l'appeler aussi *secret*, la vérification que la clé secrète ne soit jamais transmise en clair sur l'interface radiofréquence qui peut être espionnée.

- L'authentification du tag: Un lecteur doit être en mesure de vérifier un tag correct pour authentifier et identifier un tag en toute sécurité.
- L'authentification du lecteur: Un tag doit être en mesure de confirmer qu'il communique avec le lecteur correct.

D. Les étapes de vérification

Dans notre travail on a utilisé la version en ligne de la plate forme AVISPA sur le site : www.avispa-project.org après la spécification du protocole en HLPSL, *(1)* accéder au site et charger un fichier avec l'extension *.hlpsl* contenant la spécification en HLPSL, *(2)* transformer automatiquement ce fichier en une description du protocole au format IF (Format Intermédiaire). *(3)* Ce fichier qui est au format intermédiaire sera automatiquement envoyé en une entrée aux quatre outils qui vont vérifier le protocole et présenter leurs diagnostics dans un format de sortie commun. *(4)* Ces diagnostics fournissent dans le cas échéant, une trace d'attaque qui peut être visualisée graphiquement.

V. LES PROTOCOLES D'AUTHENTIFICATION DES RFID

Il existe des protocoles d'authentification des systèmes RFID utilisant des primitives et des opérateurs non supportés en HLPSL, tels que les opérateurs arithmétiques, des opérateurs logiques (et/ou), la rotation, et des fonctions de décalage, e.g. les protocoles LMAP [5] et CH [6].

Tout au long de notre papier, nous allons étudier deux protocoles d'authentification mutuelle qui utilisent des primitives cryptographiques, et qui peuvent être spécifiés en HLPSL pour qu'ils deviennent vérifiables à l'aide des outils AVISPA. Cette vérification particulière touche les transmissions sur le canal lecteur-tag seulement, car ce dernier est public, et peut subir des attaques par un intrus.

A. Le protocole FDW. :

La référence [7], représente une mise en œuvre hardware de AES pour des RFID tags avec deux protocoles simples pour l'authentification unilatérale et mutuelle. Dans cette section, on vérifie ce dernier protocole. On note que la nomination des protocoles revient aux premiers caractères de nom des auteurs (i.e. chaque caractère de mot "FDW" représente la lettre initiale du nom de l'auteur).

1) Description du protocole :

Dans ce protocole, chaque couple de lecteur `R` et tag `T` possède une clé unique et partagée `K`. Le lecteur lance le protocole par l'envoi d'un nonce `Nr` frais au tag. Le tag génère un nombre nonce `Nt` et crypte la paire `(Nt, Nr)` avec la clé partagée `K`, et l'envoie au lecteur. Le lecteur déchiffre le message en utilisant la même clé partagée et inverse l'ordre des deux nonces, crypte le message avec la même clé partagée et l'envoie au tag. La notation Alice-Bob est:

```
R → T : Nr
T → R : {Nt,Nr}_K
R → T : {Nr,Nt}_K
```

2) FDW en HLPSL :

Pour la spécification en HLPSL dans cette section, on explique les rôles de composants, et pour le principe des rôles basiques, on l'expliquera avec le deuxième protocole.

Dans le rôle `environement`, on s'intéresse au scénario de vérification suivant : deux sessions du protocole en parallèle (on la note par le symbole `/\`) concernant deux tags légitimes différents et un même lecteur légitime. Les connaissances initiales de l'intrus sont les noms des agents `t1` et `t2` qui représentent les tags, et l'agent `r` qui représente le lecteur. Cette spécification permet de détecter les attaques de type "attaque par relais" s'il existe. Les données `aut_reader` et `aut_tag` sont des constantes qui permettrent d'identifier les propriétés d'authentification du lecteur et d'authentification du tag respectivement.

La section `goal` permet de préciser les objectifs de sécurité, i.e. les propriétés, pour permettre aux outils AVISPA de faire une recherche sur les attaques.

```
Rôles Basiques
role reader ( R,T: agent, K: symmetric_key, SND,REC: channel(dy))
  played_by R  def=  local  State : nat, Nr, Nt : text
    const sec_N1 : protocol_id
    init State := 0
    transition
      1. State = 0  /\ REC(start)   =|>  State' := 1  /\ Nr' := new()
      /\ SND(Nr')  /\ witness(R,T,aut_reader,Nr')
      2. State = 1  /\ REC({Nt'.Nr}_K)   =|>   State' := 2
      /\  SND({Nr.Nt'}_K)  /\ secret(Nt',sec_N1,{R,T})
      /\ request(R,T,aut_tag,Nt')
  end role
role tag ( T,R: agent,K: symmetric_key, SND,REC: channel(dy))
  played_by T     def=
     local  State  : nat, Nt,Nr  : text
      const sec_N2 : protocol_id
     init State := 0
     transition
      1. State = 0  /\ REC(Nr')   =|>   State' := 1  /\ Nt' := new()
      /\ SND({Nt'.Nr'}_K)
      /\ secret(Nt',sec_N2,{T,R})     /\ witness(T,R,aut_tag,Nt')
      2. State = 1 /\ REC({Nr.Nt}_K)  =|>  State' := 2
      /\ request(T,R,aut_reader,Nr)
  end role
Rôles Composants
role session(T,R : agent,K : symmetric_key) def=
  local St,Rt,Sr,Rr : channel(dy)
  composition
  tag(T,R,K,St,Rt) /\ reader(R,T,K,Sr,Rr)
end role
role environment() def=
   const t1,t2,r : agent, k1,k2 : symmetric_key,
   aut_tag, aut_reader :protocol_id
   intruder_knowledge = {t1,t2,r}
   composition
   session(t1,r,k1) /\ session(t2,r,k2)
end role
Propriétés
goal
secrecy_of sec_N2, sec_N1
authentication_on aut_reader % authentification du lecteur par Nr
authentication_on aut_tag  % authentification du tag par Nt
end goal
  environment()
```

Figure 2. Spécification HLPSL du protocole FDW

3) Le résultat de la vérification :

Après la vérification de ce protocole par les outils AVISPA, le résultat est comme suit:

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfileUcVIsk.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.06s
  visitedNodes: 44 nodes
```

```
depth: 8 plies
```

Ce résultat signifié en clair qu'il n y a pas d'attaque détectée pour la confidentialité du nombre `Nt` (vérifié par `sec_N1 et sec_N2`), ou pour l'authentification du tag ou l'authentification du lecteur. On peut ainsi déduire que le diagnostic de la plateforme AVISPA pour ce protocole est *sûr*.

B. Le protocole HMNB

La référence [8] représente le protocole HMNB, ce dernier utilise la primitive cryptographique qui s'appelle la fonction de hachage.

1) Description du protocole

Le protocole est lancé par le lecteur, tel que le lecteur génère un nonce `Nr` et l'envoie à tag. Le tag génère un nonce `Nt`, la réponse du tag dépend de la valeur de `S`. Dans le cas où le processus se termine avec succès et aucun des messages n'est bloqué ou perdu, la valeur de `S` est égale à 0. Dans le cas contraire, la valeur de `S` vaut 1, ce cas devrait se produire rarement. La figure 3 décrit ce protocole.

On propose de modéliser ce protocole dans le cas `S=0`. La mise à jour est faite avant la dernière transition au niveau du lecteur après la transition au niveau du tag, dans ce cas `IDP` égal la valeur initiale de `ID` avant sa mise à jour, et nouveau `ID` égal `H(ID,Nr)`. La notation Alice-Bob proposée est la suivante :

```
R → T:Nr
T → R:H(ID),Nt
R → T: H(IDP,Nt)   % tel que IDP' :=ID
                   % et ID' := H(ID,Nr)
```

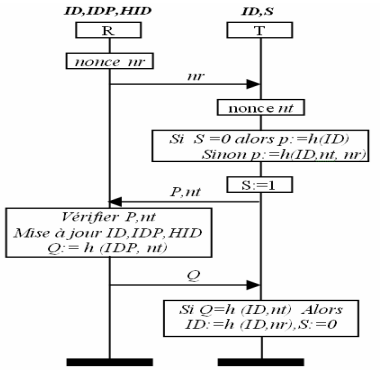

Figure 3. Le protocole HMNB

2) HMNB en HLPSL:

Dans cette section on détaille les rôles basiques de la spécification du HMNB en langage de spécification HLPSL, pour les autres rôles, il reste le même principe avec le protocole FDW qui est déjà expliqué dans la *section A.2*, sauf que des constantes et des variables utilisées changent. Ici `K, k1, k2` remplacés par `ID`, `id1`, et `id2` respectivement. Et ajoute la variable `H` dans les paramètres de `session`.

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

```
role reader ( R,T: agent, ID : text,
              H : hash_func,
              SND,REC: channel(dy)) played_by R def=
      local  State : nat,
             Nr,Nt : text,
             IDP: hash(text)
const sec_IDP : protocol_id
  init State := 0
                          . . .
role tag ( T,R: agent,
              ID : text,
              H : hash_func,
              SND,REC: channel(dy)) played_by T def=
      local  State  : nat,
             Nt, Nr : text,
             IDP    : hash(text)
      const sec_ID : protocol_id
 init State := 0
              . . .
```

Ici nous avons :

- `T, R` : des agents jouant les rôles `tag` et `reader` respectivement.
- `SND, REC` : deux canaux pour l'émission et la réception des messages respectant le modèle Dolev & Yao.
- `sec_ID, sec_IDP` : sont des constantes permettant d'identifier la propriété à vérifier, ici la confidentialité des variables ID et IDP.
- `IDP` : contient la valeur de `ID` avant la mise à jour.
- `State := 0` : la variable State initialement instanciée par la valeur 0.
- `Agent, text, hash_fuc, channel (dy), nat, hash(text), protocol_id` : sont des types.

En HLPSL le premier caractère des noms des variables est majuscule, et pour les constantes, les types et les mots-clé, le premier caractère est minuscule.

Avant d'expliquer les transitions des rôles `reader` et `tag`, Il faut également noter la notion de présence du "*prime*" dans la fin d'un variable. Si un agent souhaite 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`.

```
0. State = 0  /\ REC(start) =|> State' := 1
   /\ Nr' := new()   /\ SND(Nr')
   /\ witness(R,T,aut_reader,Nr')
2. State = 1  /\ REC(H(ID).Nt')  =|>
         State' := 2  /\ IDP' :=ID
   /\ID' :=H(ID.Nr) /\ SND(H(IDP'.Nt'))
   /\ secret(IDP',sec_IDP,{R,T})
   /\ request(R,T,aut_tag,Nt')
end role
```

La première transition du rôle `reader` signifie si la valeur de `State` est 0 et que le message dans le canal `REC` est `start` alors : `Nr` prend une nouvelle valeur aléatoire envoyée sur le canal `SND`. Pour le prédicat `witness` qui signifie "l'agent `R` déclare qu'il veut communiquer avec `T` et que la valeur de l'instanciation de `Nr` permettra d'authentifier l'agent `T`".

Pour la deuxième transition, si la valeur de `state` est 1 et le message `H(ID).Nt'` sur le canal `REC` alors, la variable

State prend la valeur 1, et les variable IDP et ID prennent les valeurs ID et H(ID.Nr) respectivement, reader envoie le message H(IDP'.Nt) sur le canal SND. Pour le prédicat secret qui signifie "la nouvelle valeur stockée dans IDP. C'est un secret qui doit être partagé seulement entre les agents R et T". Le prédicat request qui se traduit par : "R accepte la valeur Nt' et s'appuie sur la garantie que l'agent T existe et est d'accord avec R sur sa valeur".

Ici les transitions du rôle tag :

```
1. State = 0 /\ Rec(Nr')   =|>
     State' := 1 /\ Nt' := new()
   /\ Snd(H(ID).Nt')
   /\ secret(ID,sec_ID,{T,R})
   /\ witness(T,R,aut_tag,Nt')
3. State = 1     /\  Rec(H(IDP.Nt))  =|>
         State' := 2  /\ ID' := H(ID.Nr)
         /\ request(T,R,aut_reader,Nr)
```

Les transitions du rôle tag ont le même principe du travail avec les transitions exprimées dans les paragraphes précédents, la différence est dans les messages transfères et les paramètres des primitives witness et request.

3) Le résultat de la vérification :

Les outils AVISPA détectent deux traces d'attaque sur l'authentification du tag. La figure 4 montre la trace d'attaque du protocole HMNB avec l'outil CL-Atse. Dans ce résultat d'attaque, i représente l'intrus, (r,4) le lecteur, et (t1,3) le tag. La signification des informations affichées telles que : Nr(1)et n5(Nr) des instances du nonce Nr. Nt(6)et n1(Nt) des instances du nonce Nt.

Les remarques tirées qui peuvent être à partir de la trace sont :

- l'existence de deux phases de communication. La première est entre les agents i et t1, quant à la deuxième phase, elle située est entre les agents i et r, d'où l'exclusion définitive de la communication avec le tag légitime.
- h(id1) qui est émis par l'agent i à l'agent r dans la première phase est lui-même qui est émis par l'agent t1 à l'intrus i dans la deuxième phase.

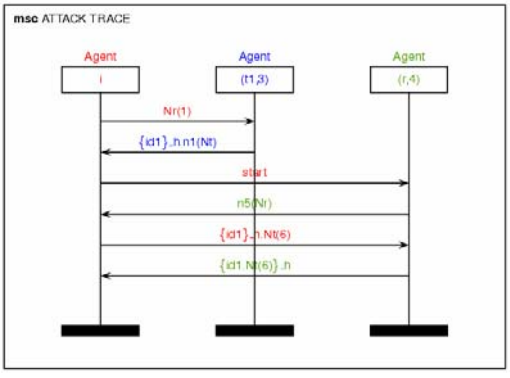

Figure 4. Trace d'attaque sur le protocole HMNB (CL-ATSE).

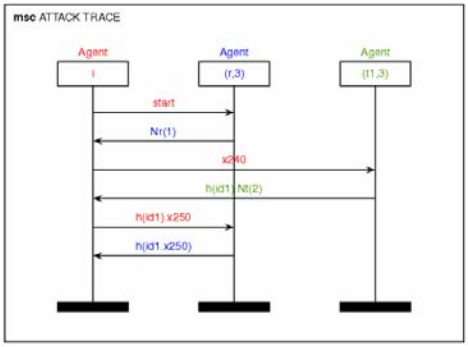

Figure 5. Trace d'attaque sur le protocole HMNB (OFMC).

La Fig. 5 illustre la trace d'attaque du protocole HMNB avec l'outil OFMC, tel que(r,3) représente le lecteur, x240 et x250 sont des variables qui ont une relation avec le travail interne de l'outil OFMC.

VI. ANALYSE DES RESULTATS

Dans cette section, on propose d'analyser les résultats en ce concerne la sécurité de protocoles. Cette analyse est basée sur les vérifications automatiques de la validation des propriétés d'authentification et de confidentialité de chaque protocole d'authentification mutuelle étudié, on résume les résultats d'expérimentation dans la table II.

TABLE II. EXPERIMENTATION SUR AVISPA

Protocole	Confidentialité		Authentification	
	Clé	*Résultat*	*Tag*	*Lecteur*
FDW	K	S	S	S
HMNB	ID	S	A	S

Notation : S: Sûr, A: Détection d'une attaque

Pour la confidentialité des informations échangées, il est à considérer qu'elle est *sécrète* pendant la transmission entre le tag et le lecteur. Pour le protocole FDW, le nonce Nt est secret par le cryptage avec la clé symétrique K connue seulement par le tag et le lecteur, ce qui implique que la clé K est *secrète*. Pour le protocole HMNB, les informations confidentielles (ID, IDP) sont cryptées par la primitive cryptographique : la fonction de hachage, qui est efficace parce que il est très difficile de trouver la valeur de ID à partir de H(ID).

Pour l'authentification du tag, L'attaque détectée dans le protocole HMNB est appelée attaque du *spoofing* ou attaque d'usurpation. Dans ce type d'attaque, un adversaire personnifie un tag RFID valide afin de se bénéficier de ses privilèges, ici la valeur de HID. Cette usurpation exige un accès plein aux mêmes canaux de communication tel que le tag original. Ce type d'attaque détecté par les outils AVISPA et le type d'attaque découvert par T. van Deursen et S. Radomirović [9] sont les mêmes.

Pour l'authentification du lecteur, les outils AVISPA ne détectent pas des traces d'attaque, on peut ainsi déduire que les protocoles FDW et HMNB sont *sûrs* pour cette propriété.

Donc pour la validation de l'authentification, Le protocole FDW est sûr, par contre, l'autre protocole peut subir des attaques logiques. Mais il faut souligner qu'il existe dans les systèmes RFID des propriétés de sécurité un peu particulières qui ne peuvent pas être vérifiées avec des outils AVISPA, comme les propriétés : l'évite de la traçabilité malveillante *'untraceability'* [10] et la désynchronisation.

VII. COMPLEXITE DU TAG

La complexité de toutes les implémentations de primitives cryptographiques exigées devrait être la plus faible possible pour maintenir le nombre requis des portes logiques, d'où le coût du tag, i.e. complexité du tag, sera aussi faible. La table III illustre les primitives cryptographiques exigées dans le tag selon le protocole d'authentification.

TABLE III. LES PRIMITIVES EXIGEES DANS LE TAG

Protocole	FDW	HMNB
Fonction de hashage		x
PRNG	x	x
Cryptage symétrique	x	

Dans notre travail, le protocole d'authentification FDW exige le chiffrement symétrique, à ce souci, on peut présenter deux algorithmes de chiffrement à clé symétrique de catégorie block cipher qui sont déjà implémentés sur le tag RFID. Le premier algorithme est AES, sa mise en œuvre efficace a été réalisée par Feldhofer et al. [11] en utilisant environ 3400 portes logiques sous forme de blocs de taille 128 bits (avec une fréquence d'horloge maximale estimée à 80MHz et la consommation d'énergie 8.2 µA dans 100kHz). Le deuxième algorithme est implémenté sur le tag par Lim et al. [12], qui ont minimisé les portes logiques exigées jusqu'à 2,608 (avec une fréquence d'horloge maximale estimée à 125MHz et la consommation d'énergie 4.3 µA dans 100kHz). Cet algorithme est appelé HIGHT.

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

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

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

VIII. CONCLUSION

Dans notre papier, nous avons traité le problème de la vérification des protocoles d'authentification en utilisant des outils AVISPA. Cette vérification est importante pour assurer les propriétés de confidentialité et d'authentification dans les systèmes RFID; mais cela ne permettra en aucun cas de confirmer que ces protocoles sont totalement valides à cause de l'existence de propriétés particulières difficiles à vérifier d'une manière automatique.

Au long de notre travail, on a concentré la présente étude sur la vérification de la communication entre le tag et le lecteur, sans négliger la possibilité d'avoir des attaques éventuelles dans l'environnement du lecteur et de la base de données.

La prise de décision pour sélectionner un protocole dépend de: complexité, performance, et sécurité. En d'autres termes, cette décision dépend de la classe du tag lui-même et du domaine auquel il est associé. Par exemple le protocole utilisé pour identifier des animaux est tout à fait différent du protocole utilisé dans le contrôle d'accès.

REFERENCES

[1] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. H_eam, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santos Santiago, M. Turuani and L. Vigneron, "The AVISPA Tool for the automated validation of internet security protocols and applications," In K. Etessami and S. Rajamani, Eds. 17th International Conference on Computer Aided Verification, CAV'2005, vol. 3576, pp. 281-285, Edinburgh, Scotland, 2005.

[2] P. Sood and T. Sadek, "RFID – Applications Based Approach to Policy," In 29th International conference of data protection and privacy commissioners, Terra Incognita, workbook series # 8, Office of the Privacy Commissioner of Canada, pp. 6-42, Septembre 2007.

[3] The AVISPA team, "HLPSL Tutorial The Beginner's Guide to Modelling and Analysing Internet Security Protocols," Technical report, AVISPA project, June 2006.

[4] D. Dolev and A. C. Yao, "On Security of Public Key Protocols," In proceding IEEE transactions on Information Theory, vol. 29, pp. 198-208, 1983.

[5] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags," In: Proc. of 2nd Workshop on RFID Security, July 2006.

[6] H.-Y Chien, and C.-W. Huang, "A lightweight RFID protocol using substring," in: EUC, 2007, pp. 422–431.

[7] M. Feldhofer, S. Dominikus and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," Cryptographic Hardware and Embedded Systems - CHES 2004 , vol. 3165, 2004, pp. 85–140.

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

[9] J. Ha, S.J. Moon, J.M. Gonzàlez Nieto and C. Boyd, "Low-cost and strong-security RFID authentication protocol," Emerging Directions in Embedded and Ubiquitous Computing, vol. 4809, 2007, pp. 795–807.

[10] T. van Deursen, S. Mauw and S. Radomirović, "Untraceability of RFID protocols," In Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks, vol. 5019 of Lecture Notes in Computer Science, pp. 1–15, Seville, Spain, 2008. Springer.

[11] M. Feldhofer, J. Wolkerstorfer and V.Rijmen, " AES Implementation on a Grain of Sand," Information Security, IEE Proceedings, vol. 152, 2005, pp. 13–20.

[12] Y-I. Lim, J-H. Lee, Y. You, K-R. Cho, "Implementation of HIGHT cryptic circuit for RFID tag," IEICE Electronics Express, vol. 6, no. 4, 2009, p.p. 180-186.

[13] K. Yüksel, "Universal hashing for ultra-low-power cryptographic hardware applications," Master's thesis, Dept. of Electronical Engineering, Worcester Polytechnic Institute, Worcester, MA, USA, 2004.