Conception et Vérification d'un Protocole d'Authentification de Système Combiné RFID-Biométrique

Noureddine Chikouche
Dpt. Informatique,
Université de Msila,
Algérie.
ciknour28@yahoo.fr

Foudil Cherif
Laboratoire LESIA,
Université de Biskra,
Algérie.
foud_cherif@yahoo.fr

Mohamed Benmohammed
Laboratoire LIRE,
Université de Constantine,
Algérie.
ben_moh123@yahoo.com

Résumé — **L'identification par radiofréquence (RFID) et les technologies biométriques ont connu des évolutions rapides au cours des dernières années et qui sont utilisés dans plusieurs applications, tel que le contrôle d'accès. Parmi les caractéristiques importantes dans les tags des systèmes RFID on trouve la limitation des ressources (mémoire, énergie, …). L'enregistrement des données biométriques sur les tags pose le problème de la taille volumineuse de ces données par rapport aux capacités des tags. Ainsi, une des méthodes efficaces pour optimiser et protéger les données biométriques est d'utiliser la fonction de hachage. Notre travail se focalise sur la conception d'un protocole d'authentification RFID qui utilise les données biométriques et qui valide la confidentialité des données privées, l'authentification et la vie privée. On utilisera les outils AVISPA & SPAN pour vérifier l'authentification et la confidentialité.**

Mots clés : **RFID, biométrie, Protocole d'authentification, non-traçabilité.**

I. INTRODUCTION

Actuellement, le problème de contrôle d'accès est très important dans plusieurs applications. Le contrôle d'accès physique consiste à vérifier si une personne demandant d'accéder à une zone (e.g. immeuble, bureau, parking, laboratoire, …etc.), a les droits nécessaires pour le faire. Les protocoles de vérification d'identité qui permettent l'accès s'appellent les protocoles d'authentification. Ils répondent aux deux questions suivantes: « Qui suis-je ?» et « Suis-je réellement la personne que je suis en train de procéder ? ». La réponse à cette première question est basée sur la reconnaissance ou l'identification de l'utilisateur qui consiste à associer une identité à une personne, tels qu'une carte à puce ou tag RFID. Concernant la deuxième question qui s'articule sur la vérification ou l'authentification de l'utilisateur, elle donne une permission à une identité proclamée. En d'autre terme, elle consiste à identifier un utilisateur à partir d'une ou de plusieurs caractéristiques physiologiques (empreintes digitales, visage, iris, etc.), ou comportementales (signature, démarche, etc.). Ces techniques sont appelées Méthodes Biométriques [14].

Parmi les techniques et les systèmes d'identification qui ont été développés rapidement au cours des dernières années, on peut constater ceux d'identification par radiofréquence (RFID) qui sont utilisées dans des domaines divers (santé, chaîne d'approvisionnement, contrôle d'accès,…etc.). Les systèmes d'identification par radiofréquence (RFID) sont composés de trois entités : *(1)* le tag (ou l'étiquette) se constitue d'une puce et d'une antenne, *(2)* le lecteur qui communique avec le tag par des ondes radiofréquences et *(3)* le serveur (ou base de données, back-end) qui utilise les informations obtenues à partir du lecteur à des fins utiles. La caractéristique principale d'un système RFID est la limitation des ressources (la mémoire, le processeur, la consommation d'énergie, etc…), d'un autre coté, ces systèmes sont nécessaires pour assurer la sécurité dans tous les niveaux du système. La différence majeure entre une étiquette RFID et une carte à puce (sans contact) est la limitation des ressources informatiques.

Dans les systèmes RFID, plusieurs protocoles d'authentification on été développés [4,5,6,7]. La différence entre ces techniques réside dans les propriétés de sécurité réalisées et la complexité d'implémentation. La plupart de ces protocoles répondent à la première question seulement « Qui suis-je ?». Au contraire pour les systèmes à cartes à puce il y a plusieurs protocoles d'authentification basés sur la technologie biométrique, on cite ici [8,9,10].

Dans ce papier, on va présenter un protocole d'authentification basé sur la combinaison entre un système RFID et un système biométrique. On vérifie la confidentialité, l'authentification du tag et l'authentification du lecteur par les outils AVISPA&SPAN [1,2]. Le protocole conçu protège la vie privée de l'utilisateur. Pour évaluer ces performances, on le comparera avec les autres protocoles RFID et les protocoles biométriques des cartes à puce.

Ce papier est organisé comme suit: la section II présente les travaux existants. La section III présente le système et les hypothèses. La section IV présente le protocole conçu. La section V présente une vérification du protocole automatiquement. Une analyse du coté de la vie privée est ensuite présentée. La section VI présente une comparaison des performances avec des travaux existants. Nous terminons par une conclusion générale.

II. TRAVAUX EXISTANTS

Dans les protocoles utilisant l'identifiant ID, deux mécanismes sont utilisés : statique et dynamique. La caractéristique 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. Ici nous présentons principalement le mécanisme d'ID statique utilisé dans cet article.

Dans le protocole RHLS (*Randomized Hash Lock scheme*) [5], les informations transmises par le tag à chaque fois qu'il est interrogé se compose d'une valeur aléatoire *nt* et la valeur H1 = h (ID,*nt*). RHLS qui découvre deux types d'attaques : attaque par rejeu et suivi de trace.

Concernant le protocole qui est proposé par Chien et Huang (protocole CH) [4], le lecteur R et le tag T partagent des secrets *k* et *ID*. Le lancement par le lecteur qui envoie un nonce aléatoire nr. Le tag produit un nonce aléatoire nt et calcule la fonction de hachage *g*, tel que $g = h(nr \oplus nt \oplus ID)$. Cette fonction de hachage et *ID* sont utilisés comme des paramètres pour la fonction *rotate*. La valeur de *ID* est tournée, elle dépend de la valeur de *g*. Le tag calcul le xor d'ID tourné et g, avant l'envoi de la moitié gauche (*Left*) des résultants et *nt* 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(*Right*) du xor d'ID tourné et g au tag. CH qui découvre une attaque du type attaques par rejeu algébrique**,** sa cause est la fausse d'utilisation de l'opérateur algébrique xor dans les messages transmis de la fonction *g*.

Lee et al. [7] proposent un protocole amélioré pour éviter deux types d'attaque: suivi de trace et attaque d'usurpation par utilisation de différentes valeurs de la fonction de hachage *h* pendant chaque authentification. Ces objectifs sont réalisés après avoir analyse du protocole [7]. Le nombre d'opérations de calcul de la fonction *h* dans le tag est quatre, ce qui est incompatible avec les capacités de stockage et de calcul du tag. De ce faite, le calcul excessif affectera inévitablement l'efficacité du protocole.

La biométrie est largement utilisé dans les protocoles d'authentification des applications de cartes à puce [8, 9, 10]. L'utilisation de ces protocoles dans les systèmes RFID dépendra de la disponibilité des ressources informatiques (mémoire, complexité, performance,…) dans les composants des systèmes RFID et surtout le tag RFID. Le dernier protocole [10] exige le calcul de sept opérations de la fonction *h* dans les phases de login et d'authentification et exige 4ℓ comme espace de stockage dans le tag. Ce nombre de calcul et cet espace de stockage influence négativement sur l'efficacité d'un protocole RFID. Une autre difficulté concerne le traitement de *Matching.* Dans les protocoles d'authentification biométrique, cette partie est faite dans la carte à puce avec la technique *Match-on-card*.

Concernant l'implémentation matériel des systèmes combinés biométrique-RFID, nous citerons deux récents travaux. Rodrigues et *al.* [15] proposent une solution d'authentification décentralisée pour les systèmes embarqués basé sur la combinaison d'une étiquette (token) et un mécanisme d'authentification biométrique. Aboalsamh [16] propose un système compact et portable qui consiste en un capteur d'empreinte digitale de CMOS (FPC1011F1) avec le processeur d'empreinte digitale FPC2020. Un circuit RFID est intégré avec le capteur et le processeur d'empreinte digitale pour créer une carte d'identité électronique (carte e-ID). La carte e-ID stocke l'empreinte digitale de l'utilisateur autorisé.

III. LE SYSTEME ET LES HYPOTHESES

1) *Le Système*

Le système d'authentification proposé est basé sur la combinaison deux sous-systèmes : un système RFID et un système biométrique. Le système RFID se compose de : un tag ($\mathcal{T}$), un lecteur ($\mathcal{R}$) et un serveur ($\mathcal{S}$). Le système biométrique utilisé se compose de deux entités, un capteur ($\mathcal{SR}$) et un serveur ($\mathcal{S}$), voir la Figure N°1.

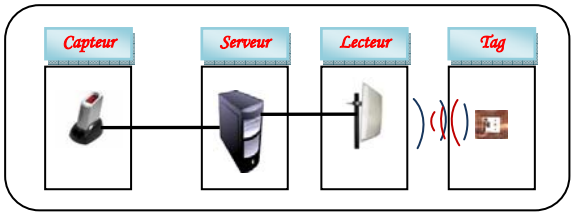

Figure 1. Systéme Combiné RFID-Biométrique

a) *Les données biométriques*

Des données biométriques peuvent être stockées sur le tag ou dans la base de données. Le template biométrique sera stocké dans le tag. Cela offre une vie privée accrue et la mobilité de l'utilisateur. Ca assure aussi que l'information sera toujours avec le détenteur de tag. Cette conception exige que le tag possède une mémoire suffisante pour stocker les données appropriées biométriques. Par exemple, la taille de l'image complète de l'empreinte exige 50 à 100 Ko, mais son template exige seulement 300 octets à 2 Ko [14]. Cette condition n'est pas toujours suffisante surtout pour les tags RFID de type passif. Dans notre système, une solution réalisable pour optimiser et protéger les données biométriques est la fonction de hachage. Cette fonction de template permet de compresser le template biométrique à une taille acceptable.

Le problème qui se pose avec les fonctions de hachage standard (e.g. SHA-1, MD5, SHA-256,…) est la comparaison entre deux templates : le template qui est sauvegardé dans le tag *h(B)* et le template qui est généré à partir de la capture *h(B')*. L'égalité $h(B) = h(B')$ pour la même personne n'est pas toujours assuré, parce que *B'* est un template dynamique où la personne ne garde jamais les même trais biométriques, (e.g. mouvement du doigt lors de l'acquisition), qui implique à l'existence d'un taux d'erreur. On citera deux travaux de recherche: Sutcu et *al.* [12] pour la modalité de visage = partir de modifications sur les fonctions SHA-1 et MD-5 (le taux d'Egal-Erreur (EER) de nouveau système est 3%). K. Kojuma [13] propose une nouvelle fonction de hachage biométrique appliquée sur les modalités empreinte et iris.

b) *Tag et Lecteur*

Le tag stocke l'identificateur (ID) et la fonction du hachage du template biométrique de la personne (HB). Cet ID est strictement confidentiel et est partagé entre la base de

données BDD du serveur ($\mathcal{S}$) et le tag. Le tag peut générer des nombres aléatoires et calcul de la fonction du hachage h d'un nombre. Les normes ISO et EPC GEN2 (*Electronic Product Code, Génération 2*) soutiennent pour produire les nonces aléatoires dans le tag. Le lecteur $\mathcal{R}$ peut génère aussi les nonces aléatoires.

c) Serveur

Le serveur a deux fonctionnalités principales :

- Pour le système biométrique : extraction des caractéristiques d'une modalité biométrique pour créer un modèle ou template ($\mathcal{B}$),
- Concernant le système RFID : il comporte la base de données qui contient la liste des identificateurs des tags ($\mathcal{ID}$).

d) Capteur

Le matériel qui permet l'acquisition d'une modalité biométrique d'une personne (empreinte digitale, visage, voix, …etc.).

e) Propriétés de sécurité exigées :

Notre protocole assure quatre propriétés: la confidentialité, l'authentification du tag, l'authentification du lecteur et la non-traçabilité.

- *La confidentialité :* la vérification que l'identificateur du tag 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 (un seul lecteur existe dans les communications entre les composants du système RFID).

  Non-traçabilité : Untraceability. Un protocole d'authentification RFID ne doit pas seulement assurer l'authentification du tag et du lecteur mais il doit aussi protéger la vie privée du tag contre le suivi de trace non autorisé: un intrus ne doit pas être capable d'obtenir n'importe quelle information de l'étiquette pour le suivi du trace.

2) Les hypothèses de l'intrus

Dans le cadre de la modélisation de protocoles d'authentification, il est nécessaire de modéliser également l'adversaire, 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* » [3]. Ce modèle est basé sur deux hypothèses importantes qui sont: le chiffrement parfait et l'adversaire est le réseau.

Le chiffrement parfait : les primitives cryptographiques sont considérées comme parfaites et sans faille, c'est-à-dire impossibles à casser. La seconde hypothèse "*l'adversaire est le réseau*" signifie que l'adversaire peut intercepter et remplacer les messages envoyés par les acteurs honnêtes du protocole, et leur envoyer des messages sous une fausse identité.

La communication entre le serveur et le lecteur et entre le serveur et le capteur est sécurisée. Contrairement à ceci, la communication entre le tag et le lecteur est quant à elle non sécurisée et basé sur les ondes radiofréquences. Notre vérification particulière (dans la section V) touche les transmissions sur le canal lecteur-tag seulement, car ce dernier peut subir des attaques par un adversaire.

IV. PROTOCOLE PROPOSE

Le Protocole proposé est divisé en deux phases: la phase d'enregistrement et la phase d'authentification mutuelle. Nous allons, par la suite, utiliser les notations suivantes:

`S, R, T`	Nom d'agent honnête (un partcipant honnête du protocole), S : Serveur ; R : lecteur ; T : Tag
`Nt, Nr`	Nonce (nombre aléatoire « frais »)
`H`	Fonction de hachage
`G`	Fonction de hachage biométrique
‖	Concaténation
`B`	Template biométrique
`ID`	Identificateur du tag partagé entre le tag et le serveur
`GB`	Fonction de hachage biométrique de B
⊕	Ou exclusif
H_D	Partie droite de la fonction H
H_G	Partie gauche de la fonction H
`X≈Y`	Signifie `X=Y±E` (`E` : taux d'erreur)

Les étapes détaillées des deux processus sont décrites ci-dessous.

1) Processus d'enrôlement

Cette phase initiale appelée aussi enrôlement. Son objectif est de créer un template (i.e. gabarit ou modèle) biométrique et stockée en liaison avec une identité déclarée (voir la figure 2.). Dans cette phase il doit exécuter les étapes suivantes pour obtenir le tag RFID.

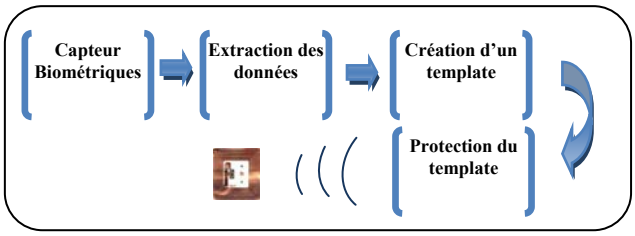

Figure 2. Processus d'enrôlement

Étape 1 : capture de la modalité biométrique de l'utilisateur autorisé, le transmettre au serveur du centre d'enregistrement ($\mathcal{CE}$).

Étape 2 : le $\mathcal{CE}$ après l'extraction des caractéristiques biométriques, crée un template biométrique B, et calcule sa fonction de hachage GB tel que GB= g(B).

Étape 3: le centre d'enregistrement stocke l'information HB et l'identificateur ID dans le tag de l'utilisateur par l'envoie au tag à travers un canal sûr.

$$\mathcal{CE} \xrightarrow{ID,\ GB} \mathcal{T}$$

2) Processus d'authentification mutuelle

Selon l'ordre des messages transmis, le processus d'authentification se déroule comme suit (voir Figure 3) :

Étape 1 : Challenge

Le lecteur RFID produit un nombre aléatoire Nr et l'envoie ensuite et 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. Dans notre protocole, le dernier cas n'est pas agrée parce qu'on exige la capture d'une seule biométrie pour chaque personne et pour chaque tag dans chaque processus d'authentification.

Étape 2 : Authentification du tag

Étape 2.1 : le tag trouvé dans *l'étape 1* génère un nonce aléatoire *Nt* et calcule $P= H_G(ID \oplus Nt \| Nr)$

Étape 2.2 : le tag envoi P avec le nonce Nt au lecteur RFID,

Étape 2.3 : le lecteur renvoy le message reçu P, Nt et le nonce Nr au serveur.

Étape 2.4 : à partir de la base de données, le serveur ca chercher un certain ID_i (tel que $1 \leq i \leq n$, n le nombre des tags) pour calcule $P_i=H_G(ID_i \oplus Nt \| Nr)$, et faire la comparaison suivante :

$$P_i=P$$

S'il est trouvé, le tag passe l'authentification du tag et est considéré comme légitime, sinon terminer.

Étape 3 : Authentification du lecteur

Étape 3.1 : le serveur calcule et envoi au lecteur Q;

$$Q = H_D(ID_i \| Nt \| N) \quad \text{Tel qu'} ID_i = ID$$

Étape 3.2 : le lecteur envoi le message Q au tag.

Étape 3.3 : Le tag calcule $H_D(ID \oplus Nt \| Nr)$ et vérifie si

$$Q = H_D(ID \oplus Nt \| Nr)$$

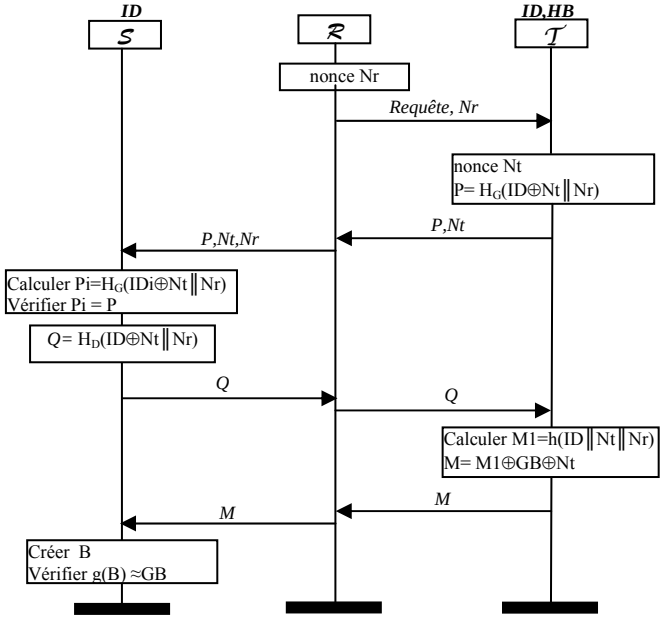

Figure 3. Protocole proposé

S'ils sont égaux, l'authentification du lecteur est réussie; sinon l'authentification du lecteur a échoue.

Étape 4 : Vérification de la biométrie

Étape 4.1 : le tag calcule M1= h(ID‖Nt‖Nr) et fait l'opération ou-exclusif de M1 avec GB et Nt. Le message résultant est M= M2⊕GB⊕Nt.

Étape 4.2 : le tag envoi M au lecteur RFID, et le lecteur renvoyé ce message reçu au serveur.

Étape 4.3 : après acquisition de la biométrie de l'utilisateur à partir du capteur il l'envoi au serveur. Le serveur extrait les caractéristiques biométriques et génère le template B. le serveur calcule la fonction de hachage du template $g(B)$.

Étape 4.4 : à partir de la base de données, le serveur calcule $M2=h(ID_i \| Nt \| Nr) \oplus Nt$, tel que $ID_i = ID$ (de l'étape 2.4) ,et extrait la valeur de HB a partir de :

$$M2 \oplus GB = P$$

Étape 4.5 : faire la comparaison de type 1:1 de $g(B) \approx GB$, s'il est validé, la personne est un utilisateur autorisé, sinon, le porteur du tag est illégitime, l'information d'échec sera envoyée au lecteur, le protocole est interrompu.

V. VERIFICATION DE SECURITE DU PROTOCOLE

1) Vérification automatique

Il y a plusieurs outils de vérification automatique de protocoles. Nous avons choisis les outils AVISPA (*Automated Validation of Internet Security Protocols and Applications*) [1] et SPAN (*Security Protocol Animator*)[2] pour les raisons suivantes: quatre outils sont disponibles utilisant différentes techniques de vérification (Model-checking, automates d'arbres, résolution de contraintes, Solveur SAT). Ces outils sont basés sur le même langage de spécification : le langage HLPSL (High-Level Protocol Specification Language). La plateforme AVISPA est l'analyseur qui modélise un grand nombre de protocoles (plus de 84 protocoles). Parmi ces quatre outils, deux outils OFMC et CL-ATSE qui peuvent vérifier les protocoles exigeant l'opérateur ou exclusif (xor). Concernant notre protocole, on vérifie les propriétés de la confidentialité de l'identificateur *ID* (sec_id_TR et sec_id_RT respectivement), la confidentialité du template B (sec_b), l'authentification du tag (aut_tag) et l'authentification du lecteur (aut_reader). Ces propriétés sont spécifiées dans HLPSL comme suite :

```
goal
    secrecy_of sec_b, sec_id_TR, sec_id_RT
    authentication_on aut_reader
    authentication_on aut_tag
end goal
```

Concernant l'authentification, il y a deux attaques possibles: l'attaque par rejeu et l'attaque Main-in-the-Middle.

Pour cela, on utilise deux types de spécification dans le rôle `environnement` de HLPSL.

a) Attaque par rejeu

Dans l'attaque par rejeu (Replay Attack), l'adversaire peut écouter le message de réponse du tag et le lecteur. Il retransmettra le message écouté sans modification au lecteur ultérieurement.

La spécification ci-dessous du rôle environnement en HLPSL dépend du traitement de deux sessions identiques entre le même tag et le même lecteur (t et r). Ce scenario permet de détecter les attaques du type "Attaque par rejeu" s'il existe.

```
role environment() def=
const t,r : agent,
      id,b : text,
      h,g,left,right : hash_func
intruder_knowledge = {t,r,h,g,hright,hleft}
composition
session(t,r,id,b,h,g,hright,hleft) /\
session(t,r,id,b,h,g,hright,hleft)
end role
```

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

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  UNTYPED_MODEL
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\BioMRFID.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 600 states
  Reachable  : 188 states
  Translation: 0.01 seconds
  Computation: 0.02 seconds
```

Ce résultat signifie en clair qu'il n y a pas d'attaque par rejeu. On peut ainsi déduire que le diagnostic des outils AVISPA&SPAN pour ce protocole est ***sûr***.

b) Attaque de l'homme au milieu

Le scenario du rôle `environnement` ci-dessous permet de détecter les attaques du type " Attaque de l'homme au milieu " (*Main-in-the-midlle Attack*) s'il existe.

```
role environment() def=
const t,r : agent,
      id,b,idti,idri,bti,bri : text,
      h,g,hright,hleft : hash_func
intruder_knowledge = {t,r,h,g,
hright,hleft,idti,idri,bti,bri}
composition
     session(t,r,id,b,h,g,hright,hleft)
 /\  session(t,i,idti,bti,h,g,hright,hleft)
 /\  session(i,r,idri,bri,h,g,hright,hleft)
end role
```

Le résultat de la vérification avec ce scénario est le même avec le scenario *a)*. On peut ainsi déduire que ce protocole est résistant à l'attaque de l'homme au milieu.

2) Analyse de sécurité

Nous analysons maintenant les propriétés suivantes: non-traçabilité, résistance à désynchronisation et résistance à Déni de service (DOS).

a) Non-traçabilité :

Pendant chaque session d'authentification, un adversaire peut observer seulement les valeurs de (Nt, Nr, M1, P, Q), où, Nt et Nr sont des nombres aléatoires et les messages M1 et Q sont calculés la partie droite/gauche de la fonction $H(ID \oplus Nt \| Nr)$. Le message $P = H(ID \| Nt \| Nr) \oplus GB \oplus Nt$. L'adversaire ne peut pas déduire les valeurs de ID parce que la fonction $H(ID \| Nt \| Nr)$ est très efficace comme est montré dans le papier de [11]. Dans les messages `M1`, `P` et `Q`, l'adversaire ne peut pas corrélée l'`ID` et `B` parce que ces deux valeurs sont secrets et `Nt` et `Nr` sont des nombres aléatoires changés dans chaque authentification. Ainsi, un adversaire ne peut pas tracer les étiquettes.

b) Résistance à la désynchronisation :

Notre protocole appartient au mécanisme statique ID où l'identificateur du tag est fixé. Donc, dans le cas de la perte de message, défaillance d'énergie ou la perte de connexion avec le serveur pendant l'authentification, cela n'affecterait pas la base de données du serveur et ne deviendrait pas un obstacle pour le protocole.

c) Résistance à DOS :

Dans certains protocoles, la technique de varier les nonces pseudonymes est utilisé pour résister au traçage et doit synchroniser les nonces entre le serveur et les tags; autrement dit, ils sont incapables de s'authentifier. Dans notre protocole, il n'y a aucune exigence de synchronisation d'état. Donc, il peut résister à l'attaque de déni de service.

Dans la Table I suivante, une comparaison de la sécurité avec des protocoles mentionnés précédemment est donnée [4 ,5 , 6,7].

TABLE I : ANALYSE DE LA SECURITE

Protocole RFID *(statique ID)*	RLHS [5]	LCAP [6]	CH [4]	LHYC [7]	**Notre protocole**
Authentification mutuelle	+	+	+	+	+
Resistance à attaque par rejeu	-	+	-	+	+
Non-traçabilité	-	+	+	+	+
Resistance à DOS	-	-	+	+	+
Resistance à Désynchronisation	+	+	+	+	+

VI. ANALYSE DE LA PERFORMANCE

Étant donné que le coût et les ressources informatiques des tags RFID sont limités. La table II illustre le coût de calcul, l'espace de stockage et le coût de la communication entre le tag et le lecteur. Le coût de calcul est fonction du nombre d'opérations de la fonction de hachage dans les phases login et l'authentification sur la carte à puce pour les

protocoles biométriques, ainsi que du nombre d'opérations de la fonction de hachage sur le tag dans les protocoles RFID.

TABLE II : ANALYSE DE LA PERFORMANCE

Protocole		Coût de calcul Tag/Carte	Espace de Stockage	Coût de la Communication		
				R →T	T →R	Σ
RFID	[4]	*1g*	*2l*	*½l*	*1½l*	*2l*
	[5]	*1h*	*1l*	-	*2l*	*2l*
	[6]	*2h*	*1l*	*1l*	*2l*	*3l*
	[7]	*4h*	*2l*	*1l*	*2l*	*3l*
C. Puce	[8]	*4h*	*3l*	*2l*	*3l*	*5l*
	[9]	*4h*	*3l*	*2l*	*3l*	*5l*
	[10]	*3h*	*4l*	*2l*	*3l*	*5l*
Notre protocole		***2h***	***2l***	***½l***	***2½l***	***3l***

Notations : *h* – nombre d'opération de la fonction de hachage,
g – nombre d'opération du générateur de nombre aléatoire avec entrée,
l : taille par bit pour chaque variable.

- ***Coût de calcul :*** le tag utilisé dans le protocole qui proposé par Lee et *al.* (protocole LHYC)[7] et les cartes à puce des protocoles biométriques exigent un nombre important d'opérations pour la fonction de hachage. Au contraire, dans le protocole Chien et Huang [4], exige une génération de nombres aléatoires avec entrée, mais il ne faut pas oublie pas l'attaque par rejeu algébrique. Dans notre protocole, on exige deux opérations de calcul de fonction *h* dans le tag, donc ces calculs sont efficaces pour les tags RFID.
- ***Espace de stockage:*** dans les protocoles biométriques [8,9], la carte à puce exige 3*l* bit et 4*l* pour le protocole [10]. Dans notre protocole, le tag exige 2*l* bit pour stocker l'identificateur (ID) et la fonction *h* de template (HB). Par conséquent, dans la mise en œuvre des protocoles, le tag ne nécessite que 2*l* bits au maximum de la mémoire, qui est adapté à des tags à faible coût.
- ***Coût de la communication:*** Concernant notre protocole, le total des bits des messages de communication tag au lecteur est : 2½*l* et pour le message de communication lecteur-tag est : ½*l*. Par rapport aux autres protocoles des cartes à puces la performance de la communication de notre protocole est plus efficace.

On peut conclure que notre protocole est efficace et adapté aux tags RFID par rapport au reste des protocoles étudiés concernant le coût de calcul, l'espace de stockage et la communication.

VII. CONCLUSION

Nous avons proposé dans cet article un nouveau protocole d'authentification RFID qui utilise les données biométriques. Ce protocole est compatible avec les ressources informatiques des tags des systèmes RFID. Concernant le problème de la taille de données biométriques, on a appliqué la fonction de hachage sur le template biométrique, qui permet d'optimiser et de protéger ces données. Ce protocole réalise le secret des données privées, l'authentification du tag et l'authentification du lecteur. Les tests expérimentaux (avec les outils AVISPA & SPAN) l'ont prouvé. On a fait une analyse de sécurité sur l'efficacité de notre protocole pour non-traçabilité, résistance aux attaques de déni de service (DOS) et résistance à la désynchronisation.

L'avantage de notre protocole est qu'il peut être utilisé dans les applications décentralisées du moment qu'on n'a pas de besoin de base de données biométrique des utilisateurs dans le système.

REFERENCES

[1] A. Armando and all., "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] Y. Glouche, and all., "SPAN (a Security Protocol ANimator for AVISPA) version 1.6," http://www.irisa.fr/celtique/genet/span/, 2009.

[3] 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.

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

[5] S. Weis, S. Sarma, R. Rivest, and D. Engels. "Security and privacy aspects of low-cost radio frequency identification systems," In D. Hutter, and all., editors, International Conference on Security in Pervasive Computing – SPC 2003, vol. 2802 of LNCS, pp.454–469, Boppard, Germany, Springer-Verlag, March 2003.

[6] S.M. Lee, Y.J. Hwang, D.H. Lee, J.I. Lim. "Efficient Authentication for Low-Cost RFID Systems," International Conference on Computational Science and its Applications - ICCSA 2005, May 2005.

[7] Y.C. Lee, Y.C. Hsieh, P.S. You and T.C. Chen, "An Improvement on RFID Authentication Protocol with Privacy Protection," Third 2008 International Conferences on Convergence and Hybrid Information Technology, South Korea: Busan, vol.2, pp. 569–573, 2008.

[8] M.K. Khan, J. Zhang, X. Wang. "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," Chaos, Solitons and Fractals, Vol. 35, No. 3, pp. 519-524, 2008.

[9] C.T. Li and M.S. Hwang. "An efficient biometrics-based remote user authentication scheme using smart cards," Journal of Network and Computer Applications, Vol. 33, pp. 1-5, 2010.

[10] Y.W. Lai , S.-C. Chang , C. Chang." An Improved Biometrics-based User Authentication Scheme without Concurrency System," International Journal of Intelligent Information Processing Vol. 1, N° 1, Sep 2010.

[11] A. Juels and S.A. Weis. "Defining strong privacy for RFID", In Proceedings of PerCom'07, pp. 342–347, http://eprint.iacr.org/2006/137, 2007.

[12] Y. Sutcu, H-T. Sencar and N. Memon. "A Secure Biometric Authentication Scheme Based on Robust Hashing," MM-SEC'05, New York, USA, August 1–2, 2005.

[13] K. Kojuma "Biometric Hash function against Quantom adversaries", Mathematical and Engineering Methods in Computer Science (MEMICS2008), Znojmo, Czech, Nov. 14-16, 2008.

[14] SmartCard Alliance, "Smart Cards and Biometrics," available to: wwww.smartcardalliance.org , Mars 2011.

[15] Joel J.P.C. Rodrigues, F.D. Heirto and B. Vaidya "Decentralized RFID authentication Solution for embedded Systems," 4th Int. Conference on Systems and Networks Communications, IEEE, pp. 174-178, 2009.

[16] H.A. Aboalsamh. "A Potable Biometric Access device using Dedicated Fingerprint Processor", WSEAS Transaction on Computers, Issue 8, Vol. 9, pp. 878-887, August 2010.