Vulnérabilités des RFID

Monty
lymonty@gmail.com

Version Zenk-Security

Table des matières

Introduction

Publication d'un travail effectué dans les cadre de mes études.

Avant de commencer, je tiens à remercier Benjamin Martin et Gildas Avoine, de l'Université Catholique de Louvain (UCL) en Belgique, ainsi que Pascal Mérindol de l'Université de Strasbourg, pour l'aide et les conseils qu'ils m'ont apportés.

Chapitre 1

RFID

1.1 Principe de fonctionnement

RFID (pour **Radio Frequency IDentification**) est une technologie d'identification par radiofréquence.
Une architecture RFID se compose en deux parties :

- Le lecteur
- Le(s) étiquette(s) (Tag(s) en anglais)

À cela peut se rajouter une base de données pour gérer les différentes identités et autorisations.

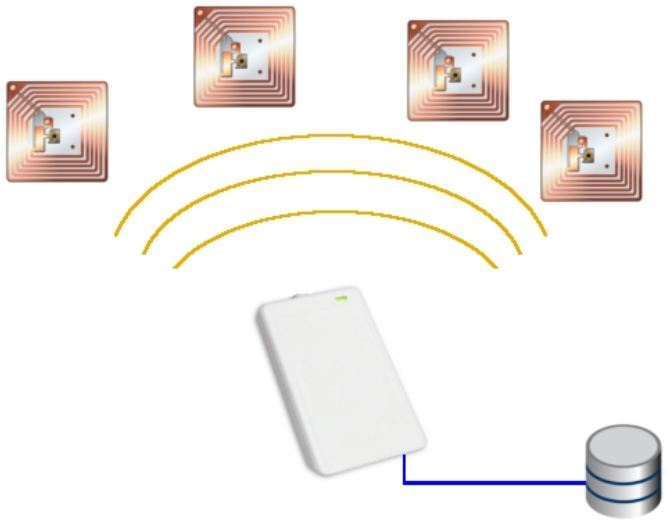

FIGURE 1.1 – Représentation d'une architecture RFID

Une étiquette est en général de petite taille. Elle est composée d'une simple puce, d'une antenne, ainsi que d'une enveloppe.
Le lecteur et les étiquettes communiquent entre eux par radiofréquence.
Les systèmes RFID sont donc des systèmes qui permettent la communication sans contact entre les étiquettes et le lecteur.

À la différence de système sans contact comme le Bluetooth ou le Wifi, les systèmes RFID sont généralement des systèmes consommant peu d'énergie, avec un faible prix de fabrication.
En contrepartie, ces systèmes permettent d'échanger une faible quantité de donnée.
Ils sont donc mis en oeuvre dans les cas où l'information à transmettre est faible, et où l'autonomie et le prix

sont des critères importants.

La fréquence utilisée varie en fonction de l'utilisation, cependant les plus rencontrées sont :

- 125 kHz (LF)
- 13,56 MHz (HF)
- 868 MHz (UHF)
- 2,45 GHz (MW)

De plus le lecteur fourni une partie ou l'ensemble de l'énergie nécessaire aux étiquettes pour fonctionner. On distingue ainsi trois types d'étiquettes :

- Les étiquettes passives : elles ne possèdent aucune batterie et sont intégralement alimentées par le lecteur.
- Les étiquettes actives : elles possèdent une batterie et peuvent émettre d'elles-même un signal.
- Les étiquettes semi-actives : elles possèdent une batterie, cependant elles ne s'en servent pas pour la communication avec le lecteur mais uniquement pour effectuer des calculs plus puissants.

Actuellement, ce sont les étiquettes passives qui sont le plus répandues, pour plusieurs raisons.
Leur fabrication étant plus simple, le prix de ce type d'étiquette est plus faible.
Ce type d'alimentation a l'avantage de permettre aux étiquettes d'avoir une durée de vie plus que conséquente (dizaine d'années). La carte étant uniquement alimentée par le lecteur, si aucun lecteur ne se trouve à proximité, elle n'émet rien, ce qui n'est pas le cas des étiquettes actives par exemple, qui doivent gérer la notion de mise en sommeil, pour ne pas user trop rapidement leurs batteries, ainsi que pour ne pas surcharger la pollution électromagnétique.
Ce sont cependant les étiquettes actives qui possèdent le plus de possibilité à l'utilisation. Mais au vu des usages adoptés, la tendance générale est plutôt vers une utilisation de système simple, où les étiquettes de type passive sont suffisantes.

1.2 Communication RFID

De base, on distingue 2 types possibles de fonctionnement pour une communication RFID entre un lecteur et une étiquette :

- Par identification : L'étiquette renvoie son ID lors de la réception d'une requête.

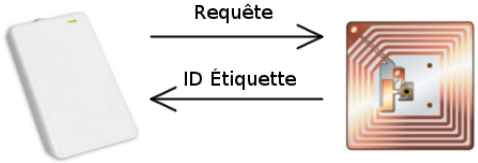

FIGURE 1.2 – Communication par identification

- Par authentification : Avant d'échanger des informations sensibles, le lecteur envoie un challenge à l'étiquette, cette dernière répond à l'aide d'une clée K et d'une fonction f partagées avec le lecteur.

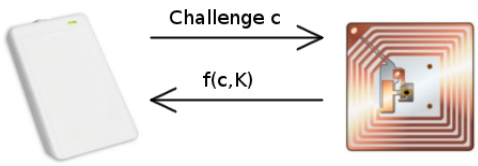

FIGURE 1.3 – Communication par authentification

Dans les faits, les protocoles de communication intégrent d'autres notions (par exemple la gestion de plusieurs étiquettes RFID à portée du lecteur) et les communications par authentifications sont plus complexes.
Des standards pour décrire ces protocoles de communications ont donc été créé, on peut citer par exemple :

- ISO 14443A 1-4 (utilisé en partie par NXP)
- ISO 14443B 1-4 (utilisé par Motorola/Atmel)
- ISO 15693-3 (utilisé par Texas Instruments)

Comme nous le verrons par la suite, la communication entre un lecteur et une étiquette, en fonction de la norme utilisée, peut s'avérer plus compliquée. Mais il est important de noter que certains systèmes RFID ne possèdent aucun mécanisme d'authentification, car leurs usages ne le requièrent pas.

1.3 Usages

Les domaines utilisant des technologies à base de RFID sont de plus en plus nombreux, on peut citer :

- Les antivols
- Les chaînes d'approvisionnements
- Le contrôle d'accès aux bâtiments
- Les cartes de transport (métro, tram)
- Le paiement sans contact
- Le passeport biométrique

FIGURE 1.4 – Exemples d'usages

Il existe plusieurs technologies qui se basent sur le RFID, par exemple :

- NFC
- Les technologies d'accès et de démarrage de voiture sans contact [4]

Ainsi les technologies RFID nous entourent déjà, sans que l'on ne le sache forcément.

On voit de plus en plus des puces sous-cutanées chez les animaux, afin d'identifier plus rapidement le bétail par exemple.
Les puces sous-cutanées chez l'Homme commencent aussi à faire leur apparition ces dernières années. Par exemple la discothèque "Baja Beach Clu" à Barcelone avait fait parler d'elle en 2004 en proposant à ses clients réguliers l'implantation d'une puce RFID (VeriShip). Cette puce leur permettant d'être identifié et de

rentrer dans la discothèque ainsi que de payer sans utiliser de porte monnaie.
Les puces sous-cutanées peuvent aussi servir à des fins médicales, ainsi aux États-Unis VeriShip commercialise la puce "Health Link" qui est une puce RFID sous-cutanée. Cette dernière permet à un médecin d'accéder au dossier médical du patient rapidement, et même si ce dernier n'est pas conscient.

Les systèmes RFID s'avèrent être aussi particulièremment efficaces pour le suivi d'objets.
Ainsi on peut imaginer de nombreuses applications de transport et de logistique à la technologie RFID. Des systèmes de tracking de bus scolaires ont ainsi vus le jour, permettant de suivre les derniers points de passages de ces véhicules. Ces systèmes donnent la possibilité aux élèves et aux parents de connaître les heures d'arrivées en temps réel.

Bien que la première utilisation des systèmes RFID soit l'identification d'équipements ou de personnes, avec la possibilité de transmettre quelques données supplémentaires, on peut imaginer d'autres utilisations.
Par exemple le projet "Bin That Thinks" (INRIA/Véolia/Etineo) vise à créer un réseau de capteurs, mélangeant des étiquettes RFID actives et passives, dans le but de faciliter le tri des déchets. [1]

Bien entendu chaque usage nécessite un type différent d'étiquette RFID.
Ainsi un badge d'accès qui nécessite une faible portée aura une fréquence entre 125 kHz et 13,56 MHz avec une étiquette passive. À contrario un système de traçabilité de palettes dans une usine aura besoin d'une plus grande distance de lecture et aura généralement une fréquence de 868 MHz.

Les systèmes RFID sont donc très diversifiés et le seront encore plus dans les années à venir, les possibilités d'utilisations différentes qu'ils apportent étant très nombreuses.

1.4 Sécurisations des RFID

Aux vues des usages faits par les RFID, il paraît évident que la sécurisation de ces systèmes dans certains cas est indispensable.
En effet, en fonction de leurs utilisations, la confidentialité du contenu des étiquettes RFID peut s'avérer être essentiels. Il semble logique que le contenu d'une étiquette RFID utilisée comme carte de paiement doit rester secret. Il en va de même pour une carte d'identité, ou passeport, contenant une puce RFID. Dans le cas contraire, le vol d'argent, ou l'usurpation d'identité sont des risques bien réels.

Mais il existe d'autres risques avec cette technologie. Comme nous l'avons vu, de plus en plus d'objets possèdent une étiquette RFID. Il ne serait pas surprenant que dans un futur proche, l'ensemble des codes barres des articles d'un magasin aient été remplacés par cette technologie. Cependant, il n'est pas possible de désactiver les étiquettes, à moins de les détruire (physiquement, ou par champ magnétique). Il existe donc un risque de voir apparaître une nouvelle forme de fuite d'information non voulue. Si à la sortie d'un magasin tous vos achats possèdent une étiquette RFID non désactivée et non sécurisée, n'importe qui avec un lecteur peut savoir ce que vous venez d'acheter, avec les problèmes sur la vie privée que cela entraînent.

De plus si un grand nombre d'objets que nous portons possèdent une étiquette RFID encore active, il serait possible pour une personne mal intentionnée et ayant l'équipement nécessaire, de suivre "à la trace" les endroits par ou nous sommes passés. En installant des lecteurs à des endroits stratégiques, il serait possible de récupérer le chemin emprunter par un ou plusieurs individus.

Ainsi assurer la sécurité de ces systèmes est primordial, et ce même si de premier abord le système ne semble pas contenir des informations confidentielles.

1. http://binthatthink.inria.fr

Chapitre 2

Sécurité

Comme nous l'avons vu précédemment il existe deux types de communication RFID.
Si le système mis en place fonctionne par identification, on peut considérer le système comme non sécurisé. En effet une simple interrogation de l'étiquette nous donnera son ID qu'il sera facile de reproduire par la suite.
Bien qu'évident, des constructeurs continuent à mettre en place des systèmes RFID par identification, avec les risques que cela comporte. On peut par exemple citer la présentation de Renaud Lifchitz qui durant le HES 2012 a montré qu'il était très facile de lire le contenu des cartes bancaires MasterCard et Visa compatible NFC. [1]

Dans une communication par authentification la sécurité repose, de premier abord sur le chiffrement.
Dans la majorité des cas, c'est une fonction de chiffrement symètrique qui est utilisée, ce qui permet de concevoir des étiquettes à des prix plus bas.
Du fait de leur faible puissance de calcul et de leur faible espace, il paraît évident que le premier vecteur d'attaque se trouve dans la fonction de chiffrement utilisée et dans la taille de la clée utilisée.

Cependant, comme nous allons le voir, de part de leur architecture particulière, les systèmes RFID sont vulnérables à un autre type d'attaque, qui est indépendant de la fonction de chiffrement, ou de la taille de clée utilisée.

La suite de ce document se concentrera sur ce type d'attaque, communément appelé "attaque par relais", ainsi que les variantes et les mécanismes de protections qui en découlent.

2.1 Attaque par relais

L'attaque par relais est une attaque d'homme du milieu (*man-in-the-middle*).
L'attaque consiste à faire communiquer le lecteur avec une étiquette truquée qui est reliée (à distance) avec un lecteur truqué qui se trouve à proximité d'une vraie étiquette.

FIGURE 2.1 – Schéma d'une attaque par relais

1. **Hacking the NFC credit cards for fun and debit** Renaud Lifchitz - Hackito Ergo Sum 2012

Ainsi le lecteur pensera être en communication avec une étiquette qui ne se trouvera pourtant pas dans le champs de portée du lecteur.

En particulier, quand le lecteur demandera à l'étiquette de s'identifier avec le challenge c *(1)*, l'étiquette communiquera le challenge au lecteur truqué *(2)* qui interrogera la véritable étiquette *(3)* et transférera la réponse *(4)* à l'étiquette truquée *(5)*, qui se fera ainsi passer pour une vraie étiquette *(6)*.

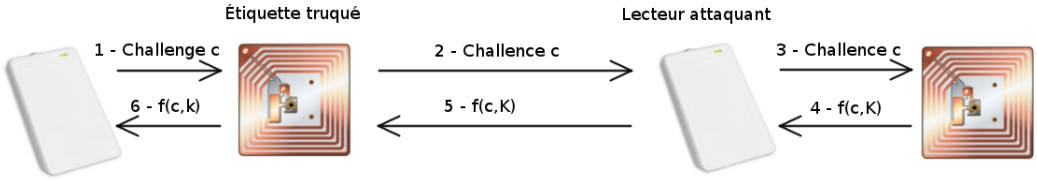

FIGURE 2.2 – Schéma détaillé d'une attaque par relais

L'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 complice 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 complice se coordonnent pour faire transmettre le challenge, et le lecteur se fera duper.

On comprend très vite que ce n'est pas le chiffrement qui est ici remis en cause, et que peu importe la fonction de chiffrement, le système est vulnérable à ce type d'attaque.

On peut penser qu'il suffit de mettre en place un système de timer pour empêcher ce type d'attaque, mais comme nous pouvons le voir dans différents travaux, cela n'est pas suffisant dans toutes les situations. [3] [4]

2.2 Distance Bounding

Pour contrer les attaques par relais que nous venons de voir, la solution actuelle consiste à mettre en place un protocole de Distance Bounding.
Le but d'un protocole de Distance Bounding est de permettre au lecteur de s'assurer que l'étiquette avec laquelle il communique se trouve dans un rayon proche de lui.

Le principe de Distance Bounding a été introduit par Stefan Brands et David Chaum en 1993 [1]. Il consiste en 3 phases d'échanges de bits, lente, rapide puis lente. En mesurant le temps d'aller-retour des échanges, le lecteur décide ou non, que l'étiquette est autorisée à continuer la communication avec lui.
La partie d'échange rapide de bits est donc primordiale, car c'est à ce moment-là que le timer autorisé doit être court, pour empêcher un attaquant d'avoir le temps de relayer le signal à un complice se trouvant hors du périmètre du lecteur.

En 2005, Hancke et Kuhn proposent une version modifiée du protocole de Brands et Chaum plus adaptée au monde des RFID [2]. Cette fois il n'y a plus que deux phases d'échange, une lente et une rapide.
La première partie sert à échanger une valeur commune, et la seconde partie permet de vérifier, en envoyant rapidement des challenges et en acceptant un faible temps de latence pour les réponses, si l'étiquette qui

répond se trouve bien à proximité.

Ainsi le vérifieur V (un lecteur) et le prouver P (une étiquette) partagent une clée secrète K et une fonction pseudo-aléatoire h dont la sortie est de taille 2n.
Premièrement V envoit un nonce N_p génèré aléatoirement à P, c'est le début de la première phase.
Puis P calcule alors h(K,N_p) et sépare le résultat en deux parties de taille n : $R^{(0)}$ et $R^{(1)}$.
La seconde phase commence alors, V envoit successivement n bits aléatoires C_i à P. À la réception de C_i, ce dernier envoit soit le i_{eme} bit de $R^{(0)}$, soit le i_{eme} bit de $R^{(1)}$, en fonction de la valeur de C_i.
Le temps d'aller-retour toléré pour l'échange entre C_i et la réponse R_i est beaucoup plus court que durant la première phase, afin de détecter une attaque par relais. Une fois les n bits envoyés, V calcul à son tour h(K,N_p) et compare les valeurs reçues avec les valeurs attendues.
C'est actuellement le protocole de référence, sur lequel se basent les nouveaux protocoles.

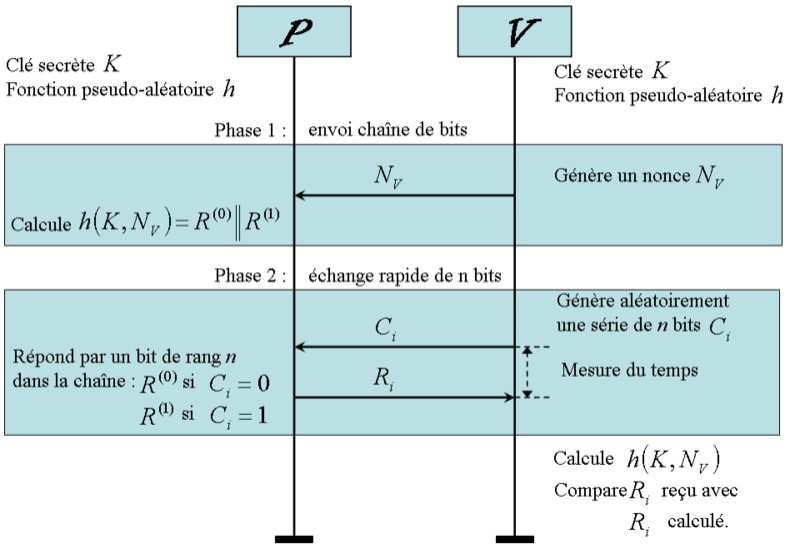

FIGURE 2.3 – Protocole de Distance Bounding de Hancke et Kuhn [2]

Cependant malgré ce type de protocole, il existe encore des attaques possibles, c'est ce que nous allons voir à présent.

2.3 Attaques sur un protocole de Distance Bounding

Il existe plusieurs façons d'attaquer un protocole de distance bounding.

L'attaque appelée *Mafia Fraud* est une attaque d'homme du milieu (*man-in-the-middle*) dont le but est de contourner un protocole de Distance Bounding et de faire croire au lecteur que l'étiquette avec qui il communique se trouve bien dans son périmètre.
C'est actuellement le type d'attaque le plus répandu.
On peut voir une Mafia Fraud comme une extension d'une attaque par relais, ou l'attaquant va souvent tenter de modifier ou prédire une partie des messages échangés.

2. source image : https://fr.wikipedia.org/wiki/Fichier:DistanceboundingprotocoledeHanckeetKuhn.png

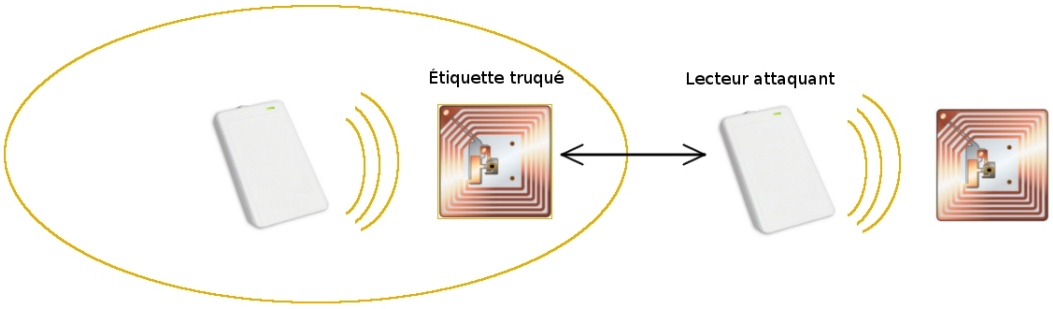

FIGURE 2.4 – Schéma d'une attaque par *Mafia Fraud*

Étant donné le protocole de Distance Bounding de Hancke et Kuhn, dans le cadre d'une attaque par *Mafia Fraud*, ou le lecteur et l'étiquette s'échangent une série de n bits lors de la phase d'échange rapide, l'attaquant a une probabilité de $(\frac{3}{4})^n$ de réussir son attaque.
En effet, si on considère la possibilité pour l'attaquant de questionner en avance l'étiquette, il peut par exemple définir l'ensemble des C_i comme égal à 0, et demander directement auprès de P, $R^{(0)}$ lors de la génération de N_p.
Si la récupération de $R^{(0)}$ se fait suffisamment vite, avant que la phase 2 ne débute, l'attaquant pourra répondre directement si $C_i = 0$ et répondra aléatoirement si $C_i = 1$.
Ainsi pour chaque bit, il aura une chance sur deux que $C_i = 0$, et donc une chance sur deux de répondre juste.
À cela s'ajoute, dans le cas ou $C_i = 1$, une chance sur deux de répondre juste avec une réponse aléatoire.
Ainsi pour un bits donné, il aura donc une probabilité de $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$.
Sur l'ensemble des n bits il aura donc une probabilité de $(\frac{3}{4})^n$ de réussir son attaque.
Il existe d'autres attaques sur les protocoles de Distance Bounding, mais en général ce sont des attaques qui reprennent le schéma d'une attaque par Mafia Fraud, avec quelques modifications/subtilités.

Par exemple l'attaque par *Terrorist Fraud* est une attaque assez similaire à une attaque par *Mafia Fraud* à la différence que l'étiquette se situant hors du périmètre lors de l'attaque n'est pas une étiquette honnête.
On peut imaginer par exemple un scénario ou une personne possède un système l'obligeant à rester dans une zone (un bracelet électronique pour les personnes sous surveillance judiciaire). Aidé d'un complice, l'individu peut mettre en place une attaque par *Terrorist Fraud* dans le but de faire croire au système qu'il se trouve toujours dans le bon périmètre.

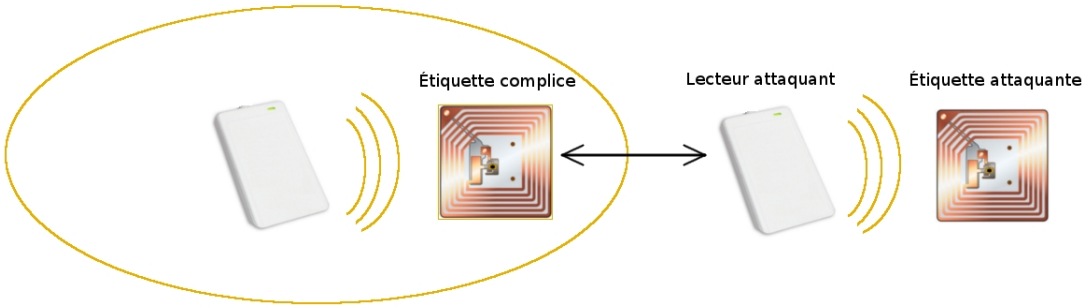

FIGURE 2.5 – Schéma d'une attaque par *Terrorist Fraud*

On peut aussi citer l'attaque appelée *Distance Fraud*, qui est une attaque effectué par un attaquant dont le but est de faire croire au lecteur que son étiquette se trouve dans son périmètre.

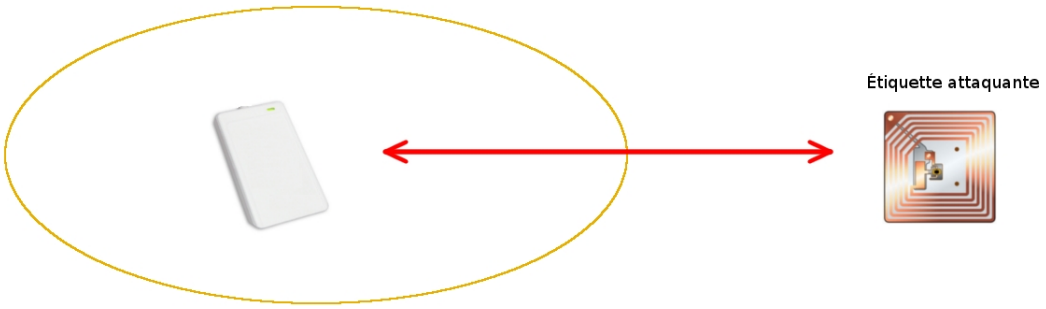

FIGURE 2.6 – Schéma d'une attaque par *Distance Fraud*

D'autres attaques existent (Distance Hijacking...). Cependant le plus grand risque de mis en pratique vient de l'attaque par *Mafia Fraud*.

2.4 Évolutions du protocole de Distance Bounding

Afin d'améliorer le protocole de Distance Bounding, et particulièrement pour augmenter sa résistance au attaque de type *Mafia Fraud*, plusieurs protocoles ont été proposés.
Il se base tous sur le principe du protocole de Hancke Kuhn, et possèdent donc une phase d'échange rapide de bits, avec un timer faible.
On peut notamment citer le protocole de Munilla et Peinado [5] qui ajoute le concept de challenge muet.
La fonction pseudo-aléatoire partagée par le vérifieur et le prouveur aura une sortie de taille 3n bits à présent.
De plus lors de la réception du nonce N_v, le prouveur enverra à son tour un nonce N_p.
La sortie de h est noté H^{3n}=P $|R^{(0)}$ $|R^{(1)}$.
P, $R^{(0)}$, $R^{(1)}$ étant tous les trois de taille n.
La challenge C_i sera considéré comme muet si P_i=0. On note P_f la probabilité que le challenge ne soit pas muet (en particulier ici P_f=$\frac{1}{2}$).
Ainsi quand le vérifieur enverra le challenge C_i en fonction de la valeur de P_i.
Il est à noter que l'on peut faire varier P_f en faisant varier la taille de h, et le nombre de bits attribué pour définir un challenge muet.
Si on définit la sortie de h comme
h(k,N_v,N_p)= H^{4n}=P^{2n} $|R^{(0)}$ $|R^{(1)}$.
On aura alors 2 bits pour définir si un challenge i est muet ou non (P_{i-1} et P_i). Ainsi si l'on veut avoir P_f=$\frac{3}{4}$ (c'est la valeur préconisée par Munilla et Peinado), il suffit de définir un challenge muet, lorsque $P_{i-1}P_i$=00.
Ainsi le challenge ne sera pas muet 3 fois sur 4 (on utise 2 bits, on a donc 4 possibilités 00,01,10,11).
En utilisant cette technique, on peut ainsi donner la probabilité que l'on veut au challenge muet.
En considérant ce protocole, l'attaquant mettant en place une attaque de type Mafia Fraud et en adoptant la même stratégie d'attaque qu'énoncé en 2.3, aura une probabilité de réussite pour chaque bit de $(P_f * \frac{3}{4})$.
Et donc pour n bits une probabilité de réussite de $(P_f * \frac{3}{4})^n$. Dans le cas ou l'attaquant répond totalement aléatoirement au challenge C_i, sans tenter de questionner à l'avance la vraie étiquette, Munilla et Peinado ont démontré qu'il aura une probabilité de réussite de $(1 - \frac{P_f}{2})^n$.
Ainsi si $P_f < \frac{4}{5}$, l'attaquant aura plus de chance de réussir son attaque en répondant de manière totalement alétoire, et si $P_f > \frac{4}{5}$, l'attaquant aura plus de chance de réussir son attaque en interrogeant le prouveur à l'avance.

Vérifieur		Prouveur
Génère N_v		Génère N_p
Envoie N_v	$\longrightarrow$	
	$\longleftarrow$	Envoie N_p
H^{3n}=h(k,N_v,N_p)		H^{3n}=h(k,N_v,N_p)
P=H_1..H_n		P=H_1..H_n
R^0=H_{n+1}..H_{2n}		R^0=H_{n+1}..H_{2n}
R^1=H_{2n+1}..H_{3n}		R^1=H_{2n+1}..H_{3n}
	pour i de 1 à n	
C_i = 0 ou 1		
Si P_i=1 Envoie C_i	$\longrightarrow$	
Sinon muet		
	$\longleftarrow$	Si P_i=1, Envoie R^{C_i}
		Si challenge ! = muet, erreur

FIGURE 2.7 – Protocole de Distance Bouding de Munilla et Peinado

Une amélioration de ce protocole a vu le jour en 2009, avec le protocole de défis mixtes de Chong Hee Kim et Gildas Avoine.[6]
Partant du principe qu'il était difficile d'implémenter le principe du challenge muet, en particulier il était difficile de synchroniser le prouveur et le vérifieur lors d'un challenge muet, ils se sont inspirés de ce dernier et ont introduit le principe du challenge mixte.
On parle de challenge mixte, car il y a deux catégories de challenge, les challenges aléatoires et les challenges prédéfinis.
La sortie de la fonction pseudo-aléatoire aura une taille de 4n bits.
On divisera donc la sortie de cette fonction en 4 parties.
On nomera les n premiers bits T, les n suivant D, et les deniers bits correspondront à $R^{(0)}$ et $R^{(1)}$.
Ainsi h(n,N_v,N_p)=H^{4n}=P |D |$R^{(0)}$ |$R^{(1)}$.
Pour chaque challenge C_i, le vérifieur enverra, si T_i=1 un nombre aléatoire, D_i sinon.
À la réception du challenge, si T_i=1, le prouveur répondra avec $R_i^{(C_i)}$, si T_i=0 et C_i=D_i, il répondra avec $R_i^{(0)}$, sinon il détectera une erreur.
Si une erreur est détectée, le protocole définit plusieurs possibilités pour le prouveur. Par exemple de répondre toujours aléatoirement lors de la réception d'une erreur, ou alors de répondre avec le bit complémentaire, d'interrompre le protocole... les avantages et inconvénients de chacune de ses possibilités sont discutés dans [6].
Chong Hee Kim et Gildas Avoine ont par ailleurs démontré que ce protocole rendait une attaque par Mafia Fraud plus difficile à réussir, avec une probabilité de réussite qui convergeait vers $(\frac{1}{2})^n$.
On constate que ce protocole donne de très bons résultats. De plus son intégration dans un système paraît plus facile à effectuer.

Vérifieur		Prouveur
Génère N_v		Génère N_p
Envoie N_v	$\longrightarrow$	
	$\longleftarrow$	Envoie N_p
H^{4n}=h(k,N_v,N_p)		H^{4n}=h(k,N_v,N_p)
P=H_1..H_n		P=H_1..H_n
D=H_{n+1}..H_{2n}		D=H_{n+1}..H_{2n}
R^0=H_{2n+1}..H_{3n}		R^0=H_{2n+1}..H_{3n}
R^1=H_{3n+1}..H_{3n}		R^1=H_{3n+1}..H_{4n}
	pour i de 1 à n	
$C_i = \begin{cases} D_i \text{ si } P_i = 0 \\ 0 \text{ ou } 1 \text{ sinon} \end{cases}$		
Envoie C_i	$\longrightarrow$	
	$\longleftarrow$	Si P_i=0 et si D_i=C_i, Envoie R^0 Si P_i=1 Envoie R^{C_i} Sinon erreur

FIGURE 2.8 – Protocole de Distance Bouding de Chong Hee Kim et Gildas Avoine

On peut aussi citer le protocole de G. Avoine et A. Tchamkerten [9].
Ce protocole se base sur une autre approche, celle des arbres binaires. De plus ce protocole sépare la partie d'authentification et la partie vérifiant la proximité du prouveur.
Tout d'abord le vérifieur envoie un nonce N_v. Le prouveur génère un nonce N_p et calcule h(K, N_v, N_p). Cette fonction a une sortie de taille de m + $2^{n+1} - 2$, m étant la taille de N_p et N_v. Il enverra alors son nonce N_p ainsi que les m premiers bits de la sortie de la fonction h. Le reste des bits servira à remplir l'arbre binaire.
La partie d'authentification est alors terminée, la partie vérifiant la proximité du prouveur commence donc. Le vérifieur choisit un nombre aléatoire pour C_i. À la réception de C_i, le prouveur renvoit la valeur correspondant au noeud se trouvant à la position C_1..C_i.

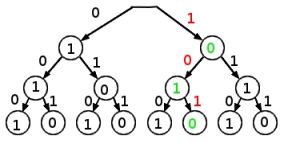

FIGURE 2.9 – Exemple d'arbre binaire pour n=3

Dans cette exemple d'arbre binaire, si le vérifieur envoie 1,0,1, le prouveur devra répondre avec 0,1,0.
Ce protocole, bien que donnant de très bons résultats, et permettant de séparer la taille des nonces (m) du nombre de tours (n), requièrent plus de mémoire et est donc plus difficile à implémenter dans la réalité. Ainsi pour n challenges la mémoire nécessaire est de $2^{n+1} - 2$

Dans une approche similaire, on peut citer le protocole Poulidor [10] développé par R. Trujillo-Rasua, B. Martin, et G. Avoine en 2010.
Ce protocole basé sur le protocole de de G. Avoine et A. Tchamkerten, vise à introduire un nouveau concept, celui des graphes. Il fonctionne selon une logique similaire au protocole vu précédemment, mais en utilisant des graphes. La mémoire nécessaire pour mettre en place ce protocole s'avère plus faible. En effet, pour n challenges, la mémoire nécessaire pour représenter le graphe est de 4n.

Il existe de nombreux autres protocoles de Distance Bounding, plus ou moins performants. Ce sont des protocoles encore au stade de recherche, mais les systèmes RFID étant de plus en plus utilisés, on peut imaginer que le nombre d'attaques sur ces systèmes ne va faire que grimper dans les années à venir. La recherche dans des mécanismes de protection contre les attaques par relais est donc essentielle.

Chapitre 3

Mise en pratique

Lors de cette étude, une mise en pratique d'une attaque par relais sur un système NFC existant a été tentée. Le système mis en place utilise deux lecteurs ACR122u faisant office de relais, un lecteur Omnikey 5121 faisant office de vrai lecteur, ainsi que d'une carte Mifare.

3.1 NFC

On parle de NFC pour Near Field Communication.
NFC est une technologie de communication sans-fil basé sur la technologie RFID.
NFC fonctionne uniquement à 13,56 MHz et est basé sur différentes normes, dont la norme ISO 14443.

3.2 Cartes Mifare

Mifare est une marque de carte NFC appartenant à NXP. Il existe différents modèles de carte Mifare, on peut notamment citer :

- Mifare Classic (avec 1 ou 4K de mémoire)
- Mifare Ultralight
- Mifare DESFire

Chaque type de carte ayant ses spécificités et ses variantes.
Pour ce document, je me suis concentré sur l'étude des Mifare Classic 1K qui est un des modèles les plus répandu.
Par exemple le Pass Campus Alsace, ou la carte Badgeo de la Compagnie des Transports Strasbourgeois sont composés d'une carte Mifare 1K.

3.2.1 Organisation de la mémoire

La mémoire des cartes Mifare 1K est divisée en 16 secteurs de 4 blocs contenant chacun 16 octets. Le 4ème bloc de chaque secteur contient les deux clés, nommée A et B, de taille de 6 octets, ainsi que 4 octets contenant les bits d'accès (un octect par bloc du secteur).
Chaque bloc a besoin d'être authentifié, soit par la clé A, soit par la clé B, en fonction des bits d'accès. De plus les accès (lecture/écriture) sont aussi déterminés par les bits d'accès.
Le premier bloc, appelé "Manufacturer block" est un bloc particulier. Il contient l'UID de la carte (comprendre son identifiant), ainsi que des données d'usines et est en lecture seule.
Ces cartes possèdent donc 2+3*15 blocs de 16 octets disponible pour les données, soit 752 octects.

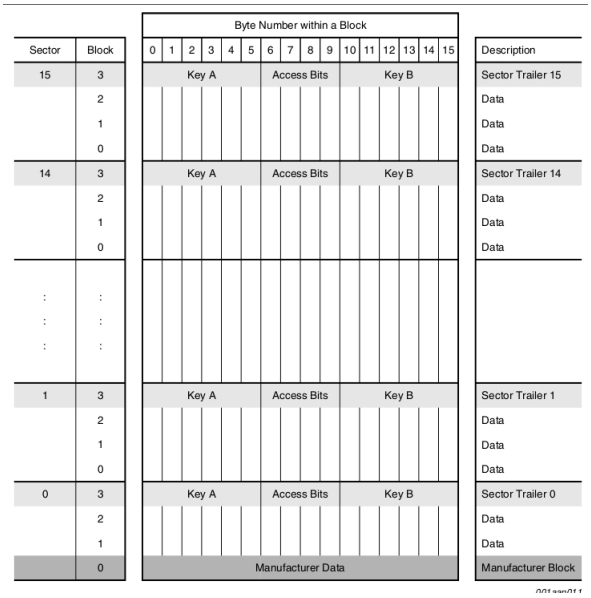

FIGURE 3.1 – Représentation de la mémoire des cartes Mifare 1K [1]

3.2.2 Communication

Les Mifare Classic 1K se basent sur les parties 1 à 3 de la norme ISO 14443A.
Les parties 1 et 2 de cette norme décrivent la partie physique ainsi que la partie traitant de la fréquence à utiliser et du traitement du signal. La partie 3, qui nous intéresse plus, décrit les protocoles d'initialisation et d'anti-collision. La partie 4, que la carte Mifare Classic 1K n'implémente donc pas, décrit les protocoles de transmission.

La partie 3 de la norme ISO 14443A établie qu'une connexion entre une étiquette et un lecteur suit ce schéma :
Tout d'abord la carte s'active lors de la réception d'une commande REQA ("REQuête" de type A), soit à la réception d'une commande WUPA ("WAke-UP" de type A) et répond avec la commande ATQA ("Answer To reQuest" de type A). La commande WUPA sera utilisée dans le cas où le lecteur veut réveiller une carte qu'il a endormi lors d'une précédente transaction avec la commande Halt.
Le protocole d'anti-collision commence alors, son but est de permettre au lecteur de selectionner une carte particulière quand plusieurs sont à sa portée.
Le lecteur envoit la commande Select. À la réception du select, l'étiquette répond en envoyant son UID. Le lecteur répond ensuite avec la commande Select Card (contenant l'UID de la carte a séléctionnée). Puis la carte répond avec la commande SAK (Select AcKnowledge). Le protocole d'anti-collision est terminé.
À la fin du protocole d'anti-collision, le lecteur peut choisir d'endormir la carte, ou de continuer le dialogue.

1. source image : MF1S503x, Product data sheet of NXP

Lecteur		Mifare 1K
REQA|WUPA	⟶	
	⟵	ATQA
Select	⟶	
	⟵	UID
Select Card	⟶	
	⟵	SAK

FIGURE 3.2 – Protocole d'anti-collision

Les valeurs de la commmande ATQA et SAK permettront au lecteur de savoir à quel type de carte il s'adresse.
Par exemple si ATQA=0x00 0x04, et SAK=0x08, le lecteur s'aura qu'il s'adresse à une carte Mifare 1K. Si ATQA=0x00 0x04 et SAK=0x09, cela correspondra à une Mifare Mini. Si ATQA= 0x03 0x04 et SAK=0x28, la carte sera une carte IBM de type JCOP31...

Send bits : 26	/REQA
Received bits : 04 00	/ATQA
Send bits : 93 20	/Select
Received bits : **52 d3 0f 27** a9	/UID Card (**52 d3 0f 27**)
Send bits : 93 70 **52 d3 0f 27** a9 f4 7b	/Select Card
Received bits : 08 b6 dd	/SAK
Send bits : 50 00 57 cd	/Halt

FIGURE 3.3 – Exemple d'échange entre un lecteur et une carte Mifare Classic 1K

Il est à noter que certaines commandes (comme la réponse au Select), possèdent à la fin un mécanisme de contrôle de redondance cyclique (CRC).

Ainsi une fois la commande SAK correspondant à une carte Mifare Classic 1K, le lecteur saura qu'il ne doit pas utiliser le jeu d'instruction décrit dans la partie 4 de l'ISO 14443A, mais utiliser les commandes définies par NXP.
Les possibilités offertes par ces commandes sont les suivantes :

- Authentification d'un bloc (Authentication)
- Lecture d'un bloc (Read)
- Écriture d'un bloc (Write)
- Décrémentation d'un bloc (Decrement)
- Incrémentation d'un bloc (Increment)
- Copie d'un bloc dans un buffer (Restore)
- Écriture d'un buffer dans un bloc (Transfert)

Comme nous l'avons vu précedemment, chaque bloc a besoin d'être authentifié, avant qu'une opération puisse être opéré sur lui.
Une fois que le lecteur a envoyé la commande Authentication avec le bloc correspondant à la carte, l'authentification se passe en 3 parties (on parle de *Three pass authentication*) :

1. La carte envoit un nombre aléatoire en guise de challenge, N_{card}
2. Le lecteur calcule la réponse au challenge N_{card}, et envoit la réponse et un nombre aléatoire N_{reader}
3. La carte calcule puis envoit la réponse au challenge N_{reader}

À partir de l'étape 1 les communications sont chiffrées.

Ainsi le protocole de communication des cartes Mifare Classic 1K ne possède pas de protection particulière

contre une attaque par relais, mis à part un timer entre chaque requête et sa réponse.

3.2.3 Chiffrement

NXP a développé son propre algorithme de chiffrement, *Crypto-1*.
Il est à noter que cet algorithme a été plusieurs fois mis à mal et n'est plus considéré comme sûr [2] [3].
Ainsi avant d'être sujet à une attaque par relais, les cartes Mifare Classic 1K sont vulnérables à une attaque sur l'algorithme de chiffrement *Crypto-1*.
On peut remarquer que même si les failles de cet algorithme sont connues, que des outils pour utiliser ses failles sont diffusées publiquement [4], et que ces cartes sont déconseillées par des organismes gouvernementaux [5], elles continuent a être employées et vendues pour des usages qui nécessitent un certain niveau de sécurité.

3.3 Lecteur ACR122u

Le type de lecteur choisit pour mettre en place une attaque par relais est le lecteur ACR122u de la société Advanced Card Systems Ltd.
Ce lecteur peut lire les cartes RFID fonctionnant à une fréquence de 13.56 MHz, il supporte donc les cartes NFC, dont les cartes Mifare Classic 1K.
En plus de son faible prix, ce lecteur a été choisit car il permet d'émuler une étiquette RFID et peut donc se faire passer pour une carte Mifare Classic 1K par exemple.
En pouvant émuler une carte, ce lecteur rend l'attaque par relais accessible à toute personne ayant des notions en informatique.
De plus ce lecteur est compatible avec les standards CCID et PC/SC, qui permettent une programmation simplifiée. Ce standard, initialement développé pour Microsoft Windows ©, a vu une implémentation libre développé pour linux, du nom de pcsc-lite, utilisant le démon pcscd.
L'ACR122u fonctionne avec une puce RFID/NFC PN532, qui est fabriquée par NXP.

3.4 Libnfc

La librairie utilisée est la libnfc [6] qui est une librairie open source écrite en C.
Cette librairie permet la lecture/écriture de cartes, l'émulation ainsi que la mise en place d'attaque par relais sur certaines cartes.
La version de la librairie utilisée est la version 1.6.0-rc1.

3.5 Tentatives d'attaques par relais

Plusieurs tentatives d'attaques par relais ont été mises en place durant cette étude, mais aucune n'a donnée de résultat satisfaisant. Nous allons à présent voir ces tentatives.
Tout d'abord, le premier essai visait à mettre en place une attaque par relais simple. L'étude du fonctionnement de la libnfc, ainsi que des documentations constructeurs, à la fois du lecteur ACR122u et des cartes Mifare 1K ont été nécessaires afin de bien comprendre tous les mécanismes en jeu.
Dans un premier temps, l'attaque consistait donc uniquement à relayer les messages entre la carte Mifare 1K et le lecteur Omnikey 5121, en se servant de deux lecteurs ACR122u en tant que relais, suivant ce schéma :

2. **Reverse-Engineering a Cryptographic RFID Tag**. Karsten Nohl, David Evans, Starbug and Henryk Plötz
3. **The MIFARE Classic Card is Hacked**. Blog of Mark Diodati
4. https://code.google.com/p/crapto1/
5. http://www.securite-informatique.gouv.fr/gp_article654.html
6. http://www.libnfc.org/

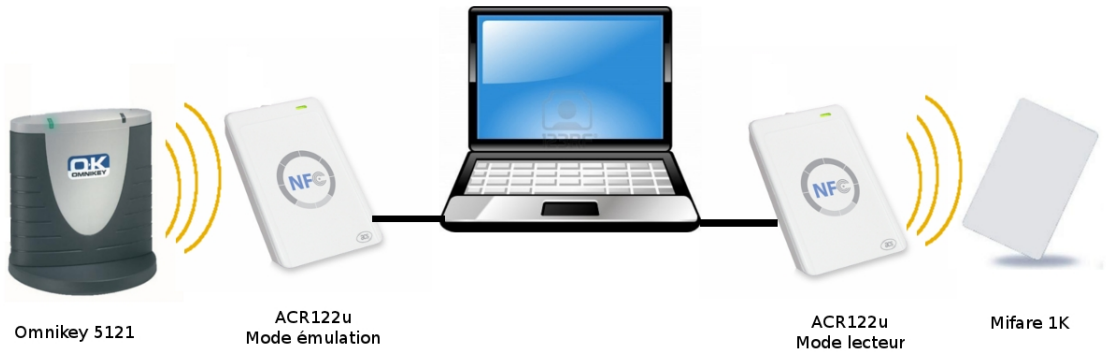

FIGURE 3.4 – Schéma de l'attaque par relais mis en place

Le relais entre les deux lecteurs ACR122u fonctionnait bien, cependant il s'est avéré que seuls le message REQA, sa réponse ATQA, et par moment la commande Select et sa réponse étaient relayés.

Send bits : 26	/REQA
Received bits : 04 00	/ATQA
Send bits : 26	/REQA
Received bits : 04 00	/ATQA
Send bits : 26	/REQA
Received bits : 04 00	/ATQA
Send bits : 93 20	/Select
Received bits : 52 d3 0f 27 a9	/UID Card
Send bits : 26	/REQA
Received bits : 04 00	/ATQA
Send bits : 26	/REQA
Received bits : 04 00	/ATQA

FIGURE 3.5 – Exemple de messages relayés durant l'attaque par relais

Il est apparu très vite que le problème venait du temps écoulé entre le moment où la requête était envoyée par le lecteur, et le moment où il reçevait sa réponse. En effet un "time-out" est déclenché lorsque ce temps est trop long. Ainsi le message REQA était bien relayé à la carte, qui répondait avec le message ATQA, mais la réponse arrivait trop tard. Il arrivait par moment que la réponse arrive à temps, le lecteur envoyait donc la commande suivante, à savoir la commande Select. La carte répondait alors bien avec son UID, mais la réponse arrivait à nouveau trop tard.
Le problème venant du fait que la partie d'émulation d'une étiquette prenait plus de temps pour dialoguer, que la partie simulation de lecteur.

À ce moment, j'ai essayé d'améliorer la réactivité du lecteur émulant la carte Mifare 1K, en modifiant la libnfc, et par exemple en enlevant des vérifications et des tests, qui n'étaient pas nécessaires dans mon schéma d'attaque, cependant l'émulation ne s'en est pas trouvée plus rapide.

Il a été tenté de passer le protocole d'anti-collision en répondant directement aux bonnes valeurs, sans relayer

les messages, et ne commencer à relayer qu'une fois le protocole d'anti-collision passé. Cette tentative a été effectuée dans le but de voir si un timer plus grand, une fois le protocole d'anti-collision passé, était présent.
Ainsi le lecteur simulant une carte Mifare Classic 1K répondra automatiquement en renvoyant les messages ATQA, UID Card, et SAK. Le lecteur étant un mode lecteur interrogera automatiquement la vraie carte avec les commandes REQA, Select et Select Card.
Une fois le protocole d'anti-collision passé, les messages seront relayés. En particulier le premier message relayé sera un message d'authentification, contenant un challenge.

Lecteur		ACR122		ACR122		Mifare 1K
REQA|WUPA	→			REQA|WUPA	→	
	←	ATQA			←	ATQA
Select	→			Select	→	
	←	UID			←	UID
Select Card	→			Select Card	→	
	←	SAK			←	SAK
message m1	→	relais	→	relais	→	
	←	relais	←	relais	←	réponse r1
message m2	→	relais	→	relais	→	
	←	relais	←	relais	←	réponse r2
...	...	...	...	..	..	..

FIGURE 3.6 – Schéma de la seconde attaque mis en place

Cependant nous obtenons les mêmes résultats et seuls quelques messages sont relayés.
On en conclut que la carte Mifare 1K a une conception qui l'a rend plus résistante à une attaque par relais. La principale difficulté vient du fait que cette carte n'est pas compatible avec la partie 4 de la norme ISO 14443. En effet cette partie rajoute la possibilité de modifier la valeur du "time-out" du lecteur, afin de demander plus de temps pour envoyer la réponse (voir la partie 7.2 "Frame waiting time" de la norme ISO 14443-4).
D'autres cartes respectent cette norme, telle que la carte Mifare DESfire.
Il est à noter qu'avec un lecteur possédant un temps de réaction plus rapide en émulation, le problème serait peut-être résolu.

3.6 Perspectives

Comme nous venons de le voir, la mise en pratique d'une attaque par relais peut s'avérer plus difficile qu'au premier abord, en fonction de la carte et du matériel à disposition.
Ainsi même si des attaques par relais ont déjà été mises en pratique avec des lecteurs ACR122u [7], généraliser l'attaque pour n'importe quelle carte est plus compliquée.
Il serait pertinent, dans une logique de continuité, d'essayer avec le même matériel de mettre en place une attaque par relais sur une carte compatible avec la 4ème partie de la norme ISO 14443A, comme la carte Mifare DESfire, et de s'aider de la possibilité d'augmenter le timer du lecteur.
Il serait aussi intéressant de mettre en place une attaque par relais, non pas avec un lecteur ACR122u, mais avec un autre matériel, comportant moins de latence dans l'exécution. Par exemple on pourrait remplacer le lecteur ACR122u en émulation, par une carte embarquée de type Arduino avec un module NFC.

7. **A.Laurie 2009**, http://rfidiot.org/

Bibliographie

[1] S. Brands et D.Chaum, *Distance-Bounding Protocols* , 1993

[2] G. Hancke et M. Kuhn, *An RFID Distance Bounding Protocol* , 2005

[3] G. Hancke, *A practical relay attack on ISO 14443 proximity cards*, 2005

[4] Aurélien Francillon, Boris Danev, Srdjan Capkun, *Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars*

[5] Jorge Munilla, Alberto Peinado, *Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels*, 2008

[6] Chong Hee Kim, Gildas Avoine, *RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks*, 2009

[7] Gildas Avoine, Muhammed Ali Bingöl, Süleyman Kardaş, Cédric Lauradoux, Benjamin Martin, *A Framework for Analyzing RFID Distance Bounding Protocols*

[8] NXP *MIFARE Classic 1K - Mainstream contactless smart card IC for fast and easy solution development*, Rev. 3.0 — 2 May 2011

[9] Gildas Avoine, Aslan Tchamkerten, *RFID Authentication Protocol : Balancing False-Acceptance Rate and Memory Requirement*, 2009

[10] Rolando Trujillo-Rasua1 , Benjamin Martin , Gildas Avoine , *The Poulidor Distance-Bounding Protocol*, 2010