La Cryptologie dans les ondes

Mounir NASR ALLAH, sous direction de Guénaël RENAULT

Mai 2014

Table des matières

Chapitre 1

Le projet

Le projet s'effectuera en monôme, sous la direction de Guénaël Renault, Maitre de Conferences à l'université Pierre et Marie Curie, membre de l'équipe POLSYS (INRIA/UPMC) du LIP6.
Etant de nature exploratoire, le projet s'inscrira dans le domaine de la sécurité des équipements RFID/NFC.
Le but étant de comprendre les mécanismes de sécurisation, et de les attaquer par différents moyens : rétro-ingénierie, cryptanalyse, attaques par canaux auxiliaires, etc ...
Les domaines de compétences requis pour mener le projet est : la cryptologie, l'architecture des systèmes, l'électronique, l'algèbre, et la programmation embarquée (Arduino, FPGA, arm).

La première partie du rapport fait référence aux aspects théoriques nécessaires à la compréhension général du projet, on y verra donc une présentation détaillée de la RFID, de la génération d'aléatoire, de la cryptographie et plus précisément du chiffrement à flot. Nous verrons ensuite l'aspect pratique du projet, on implémentera un contrôleur d'accès utilisant les carte MiFare Classic, et du code permettant de configurer cette carte. Nous attaquerons ensuite ce système d'accès grâce à la Proxmark, et nous finirons par aller un peu plus loin en étudiant d'autres types de transpodeurs.

Chapitre 2

Présentation de la RFID

2.1 La RFID

La mission de la RFID est de permettre d'identifier les objets ou les humains qui la porte, dans certaines applications, les risques d'usurpation d'identité sont faibles, dans d'autres il faut garantir l'authenticité de l'identification.
L'identification par radiofréquence, est devenue aujourd'hui une technologie incontournable. Cette technique qui permet d'identifier à distance des objets, des animaux ou des personnes sans contact physique ni visuel, est relativement simple à mettre en œuvre.
Ainsi dans certaines applications, le risque d'usurpation d'identité et la sécurité des données sont très importantes, tandis que dans d'autres ils sont inutiles.
Nous allons donc dans ce projet étudier les différents moyens développés pour sécuriser ces systèmes.

2.1.1 Contexte

L'histoire de la RFID (Radio Frequency IDentification) nous vient d'une application militaire datant de la seconde guerre mondiale, pour identifier si les avions dans l'espace aérien étaient amis ou ennemis, un transpondeur était alors installé dans les avions afin de répondre aux interrogations des radars.
Dans le milieu des années 70, Roland Moreno fondateur de Innovatron et Michel Hugon de la société BULL déposent un brevet sur leur invention, la carte à puce.
Après cette invention, de longues discussions sur la normalisation à l'AFNOR (Association française de normalisation) et à l'ISO (Organisation internationale de normalisation) donnera naissance à la famille de normes ISO 7816-X.
En 1987, fut fondée la société Mikron, elle développa des puces et des systèmes sans contact passifs fonctionnant par télé-alimentation à 125kHz.
En 1992, la société LEGIC fut la première entreprise à introduire une "secure smart card technology plateform" sans contact pour des applications de contrôle d'accès fonctionnant à la fréquence 13,56 MHz.
Après avoir normalisé les cartes à puces à contact, l'AFNOR et l'ISO normalise

à leurs tours la nouvelle génération de cartes à puces sans contact, ce qui donnera naissance à la famille de normes ISO 14443, qui permet l'interopérabilité : n'importe quelle transpondeur peut être lu par n'importe quelle station de base et inversement. Ainsi l' ISO 14443 définit quatre couches :

— La couche 1 : Caractéristiques physiques
— La couche 2 : Radio Fréquence, alimentation et signal
— La couche 3 : Initialisation et anti-collision
— La couche 4 : Protocoles de transmission

En 1994, la société Mikron développa un produit nommé "MiFare" (Mikron FARE-collection system), une carte à puce sans contact ayant un Kilo-octet de mémoire.
En 1998, Mikron fut acquise par Philips Semiconductores, devenu depuis NXP.
La RFID/NFC est aujourd'hui partout, avec des applications multiples, utilisée dans les bibliothèques (exemple : Maths-Info Enseignement à l'UPMC, BNF sur le site François Mitterand,..), dans les supermarchés, la domotique, pour les titres de transports (Navigo), le contrôle d'accès aux résidences et aux bureaux, les passeports, et récemment pour l'accès et le démarrage des voitures, mais aussi les nouvelles cartes bancaires sans contact qui commencent à être distribués depuis cette année en France ...
Ces nouvelles applications sensibles nécessitent une sécurité accrue, nous allons donc étudier la sécurité de la carte MiFare Classic 1K, ainsi le protocole de ce projet sera identique pour n'importe quelle carte à puce sans contact ayant un cryptosystème.
Afin d'éviter toutes confusions ou abus de langage, on appellera "station de base" l'appareil qui émet constamment un champ magnétique afin de détecter des objets en RFID et qui est donc "maître" dans la communication, et par "transpondeur" (Transmetteur/Répondeur) la carte qui répondra au signal émis par la "station de base".

2.1.2 Et la NFC dans tout ça ?

Les principales différences entre ces technologies sont :

— La portée : quelques centimètres pour la NFC, et quelques centimètres jusqu'à plusieurs mètres pour la RFID.
— Le débit lors de l'envoi des données.

La NFC fonctionne avec une fréquence de 13,56 MHz alors que la RFID couvre des fréquences allant de 125 kHz à plusieurs GHz.

Ainsi le choix entre ces deux technologies se fera selon l'application.
L'argument du choix de la NFC pour les cartes bancaires est la portée, qui permet en théorie de sécuriser 'physiquement' les cartes, mais ce n'est qu'une illusion car cette distance peut être étendue grâce à une antenne plus puissante sur la 'station de base'.

La NFC est donc un sous-ensemble de la RFID.

2.2 Les différents types de transpondeur

Il existe différents types de transpondeurs qui dépendent énormément de l'utilisation que l'on veut y faire, ainsi on classe les transpondeurs principalement par leur façon d'être alimenté électriquement, leur architecture matériel, et l'interface qu'ils proposent pour la communication avec le monde extérieur.

2.3 Alimentation

2.3.1 Transpondeur Actif

Le transpondeur est appelé "actif" lorsqu'il dispose d'une alimentation électrique et peut donc initialiser la communication avec la station de base, c'est ainsi le cas des téléphones portables qui embarquent de la NFC.
Ces transpondeurs sont relativement coûteux et leur taille est contrainte par la taille de la batterie.
Ils sont généralement utilisés pour des applications nécessitant des capacités de calcul ou des distances de communication importantes.

2.3.2 Transpondeur Semi-Actif

Le transpondeur "semi-actif" est alimenté par une batterie pour alimenter seulement la partie logique du transpondeur, mais qu'il est dépendant de la télé-alimentation pour la communication avec la station de base.
Ce cas de figure est surtout utilisé dans les applications médicales, ainsi la station de base peut récupérer les mesures du système afin de visualiser et traiter les données du patient.

2.3.3 Transpondeur Passif

Le transpondeur "passif" est télé-alimenté par induction électromagnétique, ce qui permet de n'avoir aucun dispositif de pile ou de batterie sur le transpondeur.
C'est le type de transpondeur le plus répondu, sous forme de carte au format ID-1 ou sous forme de "Pass". La plupart des transpondeurs RFID sont "passifs" donc télé-alimenté, ce qui introduit plusieurs contraintes, ainsi ils doivent consommer le moins d'énergie possible ce qui limite fortement l'utilisation de la cryptographie dans les puces car elle nécessite souvent l'implémentation d'un microprocesseur, requièrent beaucoup de calculs, de ressources et d'énergie.

2.4 Architecture

2.4.1 Identificateur

Les transpondeurs les moins chers sont dotés d'une simple mémoire accessible en lecture et qui contient un identifiant unique. Ainsi lorsque le transpondeur est dans le champs électromagnétique de la station de base, il se contente de lui envoyer son identifiant, son application se trouve principalement dans la logistique et la distribution.

2.4.2 Les transpondeurs "intelligents"

Les transpondeurs "intelligents" possèdent une capacité de calcul importantes avec éventuellement un microprocesseur permettant d'utiliser de la cryptographie symétrique, un générateur pseudo-aléatoire, et de la mémoire.
Généralement ces transpondeurs respectent un des standards ISO-14443 ou ISO-15693 afin d'assurer une interopérabilité. Ces transpondeurs peuvent également disposer de protections contres les attaques physique.

2.5 Interface

2.5.1 Carte mémoire

Ce comportement permet donc de voir le transpondeur comme une simple mémoire sur laquelle on vient écrire ou lire des données. C'est le comportement utilisé pour des applications "simples" comme par exemple les contrôles d'accès. Dans ce genre d'application la partie "calculatoire" se trouve surtout au niveau du transpondeur.

2.5.2 Services

Ce comportement permet d'accéder à des "services" offerts par la carte, ce qui correspond à une utilisation de multiples applications avec le même transpondeur. C'est par exemple le cas pour les cartes bancaires, ou bien pour les cartes de transports en communs dans lesquels on peut avoir plusieurs abonnements, et où la carte doit effectuer plusieurs calculs.

Chapitre 3

Normalisations

3.1 ISO 14 443 Type A

Depuis la création de la carte à puce, les industriels se sont organisés afin de développer des normes pour garantir l'interopérabilité. Les standards les plus répandues sont les normes ISO 14443 Type A, Type B et la norme B prime.

La normes ISO 14443, permet l'interopérabilité des dispositifs RFID, il est respecte le principe d'abstraction en s'inspirant du modèle OSI. Cette norme est constituée des quatre parties, et est utilisé dans la MiFare Classic.

3.1.1 Radio Fréquence, alimentation et signal

Il est utile de faire la distinction entre la modulation et le codage de l'information. Le codage correspond à la manière de représenter l'information à transmettre. La modulation, quant à elle, est la manière dont l'information sera portée par le signal radiofréquence (variation d'amplitude, de phase ou de fréquence, combinaison de ces paramètres). Contrairement à ce que l'on peut trouver dans les systèmes de télécommunication pair à pair, les systèmes RFID prévoient des différences de modulation et de codage suivant le sens de la communication.

Dans le cas d'un signal radiofréquence modulé en amplitude avec un indice de 100% (modulation OOK On-Off Keying), il est clair que la porteuse est régulièrement coupée. Ceci implique que l'étiquette doit pouvoir « survivre » à ces coupures d'alimentation. Il faut donc prévoir un système de stockage d'énergie particulier. Avec ce type de modulation, le choix du code représentant les "1" et "0" logiques aura son importance.

Dans un système de communication, le code est la manière de représenter les informations élémentaires. Il s'agit généralement d'états logiques (bits) comme le "1" ou le "0" ou de groupes de bits (symboles).

Pour la liaison montante qui correspond à la communication de la station de base vers le transpondeur, la norme ISO 14443 Type A utilise le code de Miller modifié pour le codage des bits. Celui ci présente l'avantage d'avoir de longues périodes d'inactivité et de brefs instant pendant lesquelles se produisent des impulsions, de et laisser beaucoup de temps libre pour télé-alimenter le transpondeur. C'est la modulation d'amplitude ASK 100% (Amplitude Shift Keying) qui est utilisée.

Pour la liaison descendante qui correspond à la communication du transpondeur vers la station de base, le transpondeur qui est dans un champ magnétique communique avec la station de base, en modulant la propre charge qu'il représente. La modulation de charge permet de créer un bit générant ou non selon les souhaits une sous-porteuse.
Le codage bit codé Manchester sous porteuse (MSC) est une variante intéressante de sous codage du codage Manchester. Pendant la moitié (ou une partie) du bit Manchester, le signal composant celui-ci est modulé par une modulation de charge en tout ou rien (modulation OOK, pour On Off Keying).

3.1.2 Initialisation et anticollision

L'un des problèmes majeurs de la RFID est que comme tout dispositif sans fil, plusieurs entités peuvent se trouver dans un même champ magnétique, il faut donc un protocole afin de gérer l'initialisation des communications, et également une gestion des éventuelles collisions qui peuvent intervenir entre la station de base et les différents transpondeurs présents dans le champs.

La plupart des transpondeurs possèdent un identification unique attribué par le fondeur lors de sa fabrication, ce qui permet d'identifier de façon unique chaque transpondeur, ce qui permet par exemple de créer des systèmes tels qu'ils soit possible à une station de base de connaître uniquement les transpondeurs avec lesquels elle a le droit de communiquer. Il est également très utile pour la gestion des collisions puisqu'il pourra également être utilisé pour sélectionner le transpondeur qui sera impliqué pendant la communication.

Grâce à l'utilisation d'un codage bit de type "Manchester coded subcarrier" (MSC), la détection d'une collision bit peut être réalisée facilement de manière robuste et indépendante de la puissance du signal reçu. Cette technique permet de disposer de 3 états logiques : 0, 1, et "collisions".

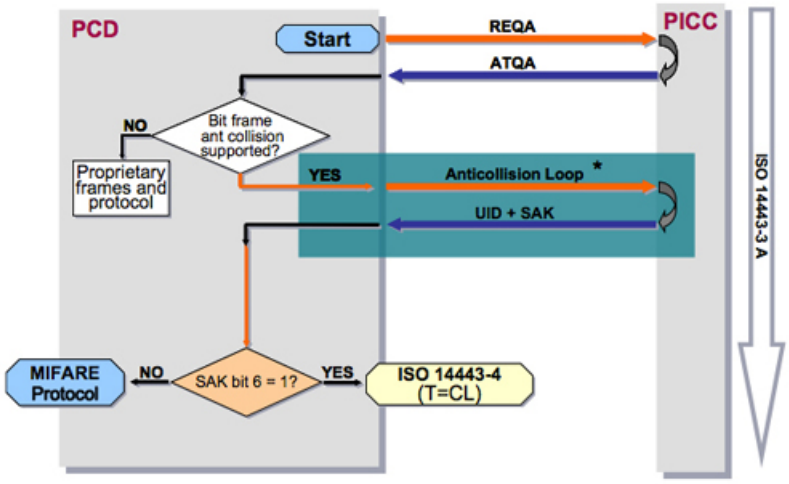

FIGURE 3.1 – Initialisation d'une communication

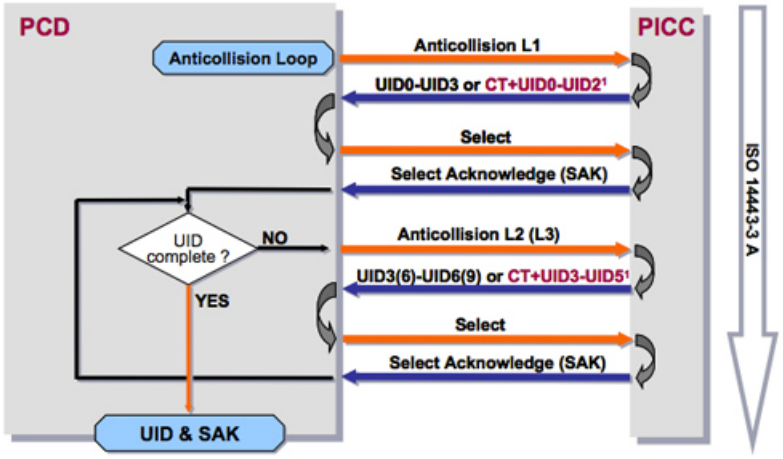

FIGURE 3.2 – Gestion des collisions

Chapitre 4

Génération d'aléa

4.1 L'aléatoire

Des nombreuses applications informatique ont besoin de créer des séquences aléatoires, malheureusement l'aléatoire "parfait" n'existe pas en informatique. Ainsi il y a plusieurs façon de créer de l'aléatoire, sur certains systèmes d'exploitation comme Linux, la génération d'aléa est une fonction qui prend comme paramètres plusieurs "mesures", tel que la température du CPU, le temps d'accès mémoire, etc... et d'en ressortir une séquence, ce qui ressemble donc plus à de la "bidouille". La génération d'aléa dépend donc souvent de l'environnement dans lequel il est utilisé.

Il est important d'avoir un générateur d'aléa cryptographiquement sûr, car si le générateur d'aléa est biaisé, alors il est facile de récupérer certaines informations, comme par exemple la clé, indépendamment de l'algorithme de chiffrement utilisé.

L'aléatoire joue donc un rôle important dans la cryptographie, car le modèle idéal que l'on appelle "cryptosystème incassable" existe et est justement basé sur la génération d'aléa parfait". Ce cryptosystème à été découvert vers 1917 par Gilbert Vernam (1890 - 1960), puis a ensuite été prouvé mathématiquement "incassable" par Claude Elwood Shannon (1916 - 2001). Ces séquences aléatoires "parfaites" doivent dans l'idéal respecter certains principes :

1) L'uniformité
Pour chaque bit généré, il y a exactement une chance sur deux d'obtenir un 1 ou 0
$p(K_i = 1) = p(K_i = 0) = \frac{1}{2}$
2) L'indépendance
Quelque soit l' des bits déjà générés pour la séquence, il est impossible de prédire quelle est l'état du prochain bit qui sera généré.

Dans la pratique, plusieurs contraintes existent : il y a toujours un compromis à faire entre vitesse de génération, et sécurité.

4.2 Le pseudo-aléatoire

Puisqu'il est très compliqué voir impossible de faire un vrai générateur d'aléa, on utilise des systèmes de génération "pseudo-aléatoire", Ces systèmes utilisent ce que l'on appelle une graine (seed), qui permettra de générer des séquences pseudo aléatoire à partir de cet état.

4.3 Les Registre à décalage à rétroaction linéaire

Les registres à décalage à rétroaction linaire, "Linear Feedback Shift Register" en anglais ou plus connu sous l'acronyme "LFSR", est un système de génération pseudo aléatoire très simple à mettre en œuvre, et très rapide. Ce système est un simple registre sur lequel on applique à chaque itération : un décalage à droite qui fait sortir un bit du registre, et qui insère le résultat d'une fonction polynômiale à la place du bit de poids fort (le bit le plus à gauche, le bit entrant).

Un LFSR se modélise mathématiquement de la sorte :

Soient :
Un LFSR sur le corps fini $\mathbb{F}_2^n$ n la taille du LFSR $E_0 = (b_0, b_1, \cdots, b_{r-2}, b_{r-1})$ l'état initial du LFSR à l'instant $t = 0$, qui est égale à la graine (seed),

Une fonction polynomiale linéaire f définit comme suit :
$f(x_0, x_1, ..., x_{r-2}, x_{r-1}) = c_0x_0+c_1x_1+...+c_{r-2}x_{r-2}+c_{r-1}x_{r-1}$ avec $(c_0, c_1, ..., c_{r-2}, c_{r-1})$, les r coefficients de connexion du LFSR dans le corps fini $\mathbb{F}_2^n$

On calcule le bit entrant b_r comme suit :
$b_r = f(b_0, b_1, \cdots, b_{r-2}, b_{r-1})$

Puis on fait sortir le bit sortant (le bit de poid le plus faible) et on fait entrer le bit entrant comme suit :
$E_r = f(b_1, b_1, \cdots, b_{r-2}, b_{r-1}, b_r)$

Les LFSR ont de bonnes propriétés d'aléa statistique mais ont de graves problèmes quand ils sont appliqués à la cryptographie, en effet il suffit de 2l+1 bits de la suite engendré par le LFSR pour retrouver la graine et donc l'état interne du LFSR, ce qui est possible grâce à l'algorithme Berlekamp-Massey.

L'un des problème principal des LFSR est que malgré leurs bonnes propriétés statistiques, ils sont linéaires, ce qui est un énorme défaut en cryptologie, de

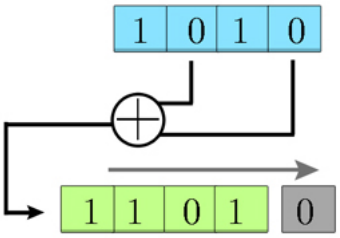

FIGURE 4.1 – Schéma d'un LFSR

part leur côté "prédictible", et qu'un attaquant pourrait ainsi modéliser le système par des équations et donc obtenir des informations grâce à leurs résolutions.

Il est donc nécessaire de combler ce défaut grâce à des fonctions booléennes, qui ont pour objectif principal de casser cette linéarité. Le choix de ses fonctions booléennes ne se fait pas au hasard, ils doivent présenter de fortes propriétés cryptographique, ce qui est assez compliqué vu l'ensemble de toutes les fonctions possibles, 2^{2^n} pour des fonctions booléennes à n entrées.

Chapitre 5

Les bases de la cryptographie

5.1 Histoire de la cryptographie

La première attestation de l'utilisation délibérée de moyens techniques permettant de chiffrer les messages date du VIème siècle avant J-C, et se nomme "scytale".
Plus tard, les armées romaine utilisèrent le chiffrement de César qui consistait à décaler l'alphabet de trois lettres, mais ce système est cryptographiquement très faible puisque vulnérable à une simple analyse statistique du nombre d'occurrence des symbôles présents dans le message chiffré, en les comparant aux statistiques des symbôles les plus fréquents dans la langue cible.

5.2 Les principes de base

Au 19ème siècles, Auguste Kerckhoffs von Nieuwenhoff (1835 - 1903) posa les principes de la cryptographie moderne. L'un des principes supposait que la sécurité d'un système de chiffrement ne résidait que dans le secret de la clé et non dans le procédé de chiffrement.
Ainsi, tous les autres paramètres doivent être supposés publiquement connus. Il est considéré aujourd'hui comme un principe fondamental par les cryptologues, et s'oppose à la sécurité par l'obscurité (Security by obscurity).
Dans la sécurité par l'obscurité on peut citer les algorithmes A3/A8 utilisés dans les GSM, et CRYPTO1 utilisé dans les MiFare Classic, tout deux cassés depuis.

C'est Claude Elwood Shannon (1916 - 2001) qui pose les bases et la rigueur mathématiques au service de la cryptographie, il introduit des outils qui permettent de mesurer et quantifier la résistance des cryptosystèmes. Ainsi en 1948 et 1949, deux articles de Shannon "A mathematical theory of communica-

tion" et "The communication theory of secrecy systems" ont donnés des assises scientifiques à la cryptographie en balayant espoirs et préjugés.

5.3 Identification et authentification

Les termes "Identification" et "Authentification" sont souvent mal utilisés dans la littérature, on désigne donc par "Identification" l'acte qui consiste à déterminer l'identité d'un objet ou d'une personne selon des caractéristiques, tandis que l'"Authentification" consiste à vérifier une identité déclarée.

Ainsi se baser seulement sur l'identification c'est prendre le risque d'être victime d'usurpation d'identité, puisque le simple fait de copier les caractéristiques décrivant un objet ou une personnes sur un autre transpondeur RFID fait de lui un "clone".

Indépendamment du système de cryptographie utilisé, on distingue deux types d'authentification : faible et forte.

5.3.1 Authentification faible

L'authentification faible consiste à ajouter à l'identifiant une valeur d'authentification calculée une fois pour toutes qui ne change pas dans le temps.
Ce mécanisme d'authentification ne protège pas d'une attaque par rejeu, et reste donc vulnérable à l'usurpation d'identité.

5.3.2 Authentification forte

L'authentification forte est un mécanisme dont la valeur d'authentification fournie par la puce n'est "jamais" la même, cette valeur est généré par la puce ou est fournie par la transpondeur et la station de base par un compteur synchronisé. Cette authentification permet de se protéger d'une attaque par rejeu, et permet donc de résoudre le problème d'usurpation d'identité.

5.4 Cryptographie asymétrique

En cryptographie asymétrique, il y a deux clés appelés clé publique et clé privée, elles ont des rôles fondamentalement différents d'où le terme "asymétrique". La clé publique est diffusé et donc connu par n'importe qui, alors que la clé privée est gardée secrète, ces clés ont des valeurs différentes, et des rôles différents. Ainsi on utilise la clé publique afin de chiffrer un message tandis que la clé privée sert à le déchiffrer.

5.5 Cryptographie symétrique

En cryptographie symétrique, les deux clés sont identiques et servent à la fois pour le chiffrement et le déchiffrement. Les clefs doivent donc être distribuées secrètement, si la clef est compromise alors l'attaquant peut écouter toutes les communications, mais également participer à la communication, l'un des problèmes majeur est que si l'on veut utiliser une clef différente pour chaque pairs d'utilisateurs, le nombre total de clefs croit très rapidement.

5.6 Chiffrement par bloc

Le chiffrement par bloc permet de chiffrer des messages clairs de taille fixe n, et produit donc des messages chiffrés de taille n. Ce chiffrement peut être utilisé à la fois en cryptographie symétrique et cryptographie asymétrique, on peut ainsi citer DES (Data Encryption Standard), 3DES (Triple-DES), AES (Advanced Encryption Standard), RSA.

On a :
La clé secrète $K = (K_1, \cdots, K_n) \in \mathbb{F}_2^n$
Le message clair $M = (M_1, \cdots, M_n) \in \mathbb{F}_2^n$
Le message chiffré $C = (C_1, \cdots, C_n) \in \mathbb{F}_2^n$

On a donc une fonction de chiffrement :

$$f_C : \mathbb{F}_2^k \times \mathbb{F}_2^n \mapsto \mathbb{F}_2^n (K, M) \mapsto C = f_C(K, M)$$

Avec : $f_C(K, M) = M_i \oplus K_i, i \in 1, ..., n$

On a également une fonction de déchiffrement :

$$f_D : \mathbb{F}_2^k \times \mathbb{F}_2^n \mapsto \mathbb{F}_2^n (K, M) \mapsto M = f_D(K, C)$$

Avec : $f_D(K, M) = C_i \oplus K_i, i \in \{1, ..., n\}$

5.7 Chiffrement par flot

Le chiffrement par flot est un fonctionnement présent exclusivement en cryptographie symétrique, ce type de chiffrement permet de chiffrer à la volée des informations de taille quelconque. Très rapide, il est surtout utilisé dans des applications où le temps et les données n'ont pas de taille fixe, on peut par exemple cité A5/1 qui a été l'algorithme utilisé pour le chiffrement des communication GSM, RC4 utilisé pour le protocole WEP, E0 utilisé pour le Bluetooth, et très présent dans les cartes à puces et transpondeurs RFID.

5.8 Système de chiffrement inconditionnelement sûr

Shanon prouva ainsi que le seul crypto-système inconditionnellement sûr est le système de Vernam (appelé one-time pad), malheureusement ce système idéal est impossible à mettre en place pour plusieurs raisons :

— La génération d'aléatoire parfait n'existe pas
— On ne doit pas réutiliser la même clef pour chaque message chiffré, ce qui oblige donc d'utiliser un autre moyen de communication pour transmettre les clefs, et introduit donc un maillon faible.

5.9 Application dans le RFID

La cryptographie asymétrique est actuellement très peu utilisé pour la RFID, à cause des ressources importantes en terme d'énergie, de coût, de porte logique, et de temps d'exécution, que demandent de tels crypto-systèmes. Malgré tout une lueur d'espoir vient d'un protocole nommé "GPS" (ISO 9798-5) pour "Girault - Poupard/Paillès - Stern", un algorithme d'authentification et de signature numérique à clé publique, basé sur le problème du logarithme discret, problème mathématique connu et utilisé à des fins cryptographique depuis des années. La rapidité de ce protocole permet une implémentations sur des composants à faible coût avec une sécurité qui peut être équivalente ou supérieure à celle de RSA.

Pour pouvoir implémenter de la cryptographie dans les cartes RFID, on a trois approches possibles :

— Implémenter des crypto-systèmes standards qui ont fait leurs preuves, ce qui peut revenir très couteux, une implémentation de l'AES avec 3600 portes logique a été réalisé.
— Modifier des crypto-systèmes standards afin de les adapter aux puces RFID et donc de les restreindre dans la façon de les utiliser.
— Concevoir de nouveaux crypto-systèmes répondant aux contraintes d'une implémentation sur puce à faible coût, mais n'ayant pas prouvés leurs résistances à des attaques.

La taille des clés nécessaire en cryptographie à clé publique pour assurer une sécurité satisfaisante est plus grande que la taille des clés en cryptographie à clé secrète. Ainsi le système RSA avec une clé de 512 bits est bien moins sûr qu'un AES avec une clé de 128 bits. Il est donc plus naturel d'utiliser des systèmes symétriques pour des applications RFID.
Les algorithmes standards ne peuvent pas être facilement implémentés dans des circuits rudimentaires d'au plus quelques milliers de portes logiques car la puce atteindrait vite des coût trop importants.

5.10 Conclusion

On dit donc d'un système de chiffrement qu'il est sûr tant qu'il n'existe aucune attaque significativement plus performante à un recherche exhaustive de la clef.

Chapitre 6

Le Chiffrement par flot

Le chiffrement par flot, stream cipher en anglais, désigne un crypto-système de chiffrement à la volée. Ainsi les messages clairs sont chiffrés au fur et à mesure de leur productions, ce genre de crypto-système est de loin le plus rapide, d'où son fort succès dans des applications à contrainte temps réel tel que la téléphonie. Le chiffrement par flot est un système symétrique, il offre un haut niveau de sécurité, car ils se rapproche du modèle de Vernam de par l'aléatoire qui y est introduit, ce qui implique lors de leurs conceptions une longue phase de tests statistiques sur le générateur aléatoire et le système complet, ainsi il existe principalement deux types de systèmes : les systèmes synchrones et les systèmes asynchrones.

6.1 Les systèmes synchrones

6.1.1 Principes

Dans les systèmes synchrones, la valeur aléatoire est généré indépendamment du message clair et du message chiffré, ce qui implique donc que l'émetteur et le destinataire soient constamment synchronisés car en cas de désynchronisation il leur est impossible de déchiffrer les messages.

On peut donc représenter ce système de cette façon :

E représente l'état du système de génération pseudo-aléatoire, avec comme initialisation E_0 l'état initial qui est généralement des bits de la clé secrète maître. Cette fonction dépend donc de l'état précédent du système.
$E_{i+1} = f(E_i, Km)$

On a Ks_i qui représente la "Key Stream", qui est une sous clef généré via la clé maître, la fonction g produit une valeur pseudo-aléatoire grâce à l'état du système et la clé maître Km.

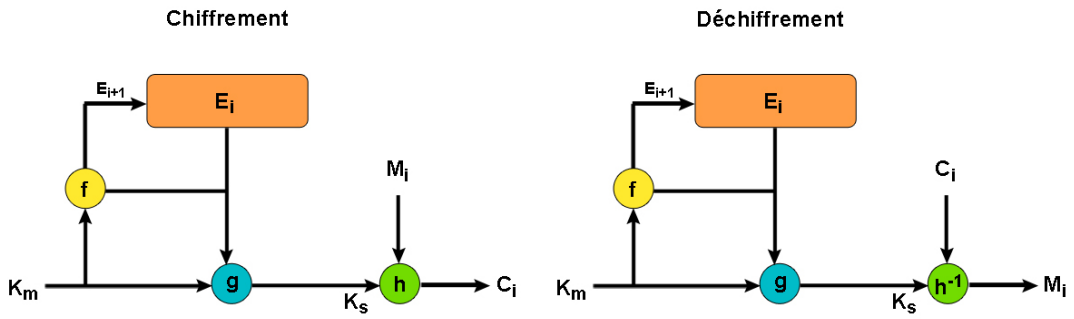

FIGURE 6.1 – Schéma d'un système synchrone

$Ks_{i+1} = g(E_{i+1}, Km)$

On a enfin les fonctions de chiffrement h de déchiffrement h^{-1} permettant de chiffrer un message clair M en un chiffré C grâce à la clé généré (Key Stream) :
$C_i = h(M_i, KS_i)$
$M_i = h^{-1}(C_i, KS_i)$

Ce qui donne schématiquement :

6.1.2 Système de chiffrement binaire additif

On appelle système de chiffrement binaire additif, un système par flot synchrone dans lequel les entrées sont de nature binaire et où la fonction h est un XOR. On a donc h et h^{-1} qui sont de simples XOR.

6.2 Les systèmes asynchrones

6.2.1 Principes

Dans les systèmes asynchrones, la valeur aléatoire est généré en fonction de la clef et de bits du texte chiffré.
On peut donc représenter ce système de cette façon :

E représente l'état du système de génération pseudo-aléatoire, avec comme initialisation E_0 l'état initial qui est un état non secret.
$E_i = (C_0, C_1, \cdots, C_{i-2}, C_{i-1})$

On a KS_i qui représente la "Key Stream", qui est une sous clef généré via la clé maître, la fonction g produit une valeur pseudo-aléatoire grâce à l'état du système et la clé maître KM.

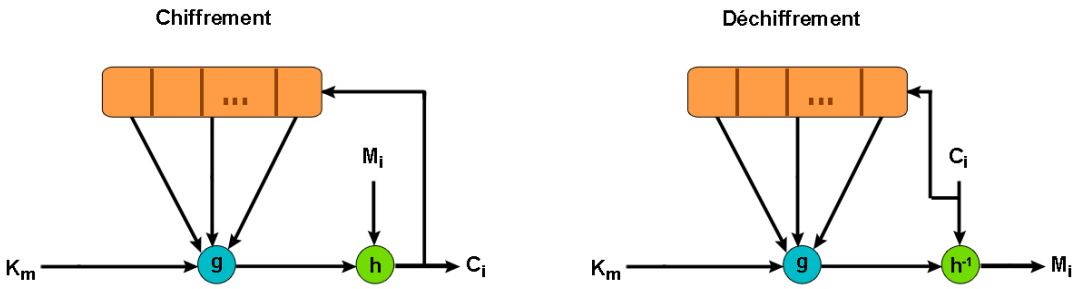

FIGURE 6.2 – Schéma d'un système asynchrone

$KS_{i+1} = g(E_{i+1}, KM)$

On a enfin les fonctions de chiffrement h de déchiffrement h^{-1} permettant de chiffrer un message clair M en un chiffré C grâce à la clé généré (Key Stream) :
$C_i = h(M_i, KS_i)$
$M_i = h^{-1}(C_i, KS_i)$

Ce qui donne schématiquement :

Chapitre 7

Attaque d'un système sécurisé

L'attaque d'un système peut se faire par différents moyens, on parle alors d'attaques logiques ou d'attaques physiques.
Les attaques logiques sont généralement des attaques sur le logiciel embarqué dans le système, qui peut dans certains cas faire fuir certaines informations sensibles.

Les attaques physiques sont des attaques sur le matériel, ainsi on différencie différents types d'attaques physiques :

— Le attaques invasives
— Les attaques passives
— Les attaque actives

7.1 Le attaques invasives

Ce type d'attaque nécessite du matériel professionnel, c'est le type d'attaque le plus couteux en temps et en budget. Le principe de ces attaques est soit de faire de la rétro-ingénieurie en découpant chaque couches de la puce, les prendre en photo pour en faire une analyse bottom/up grâce à des logiciels professionnels afin de comprendre l'architecture de la puce, soit en créant de nouvelles connexions directement sur des fils de la puce afin d'injecter ou d'extraire des signaux. L'inconvénient de ce type d'attaque est que la puce peut être ensuite inutilisable à cause des dommages qui peuvent être engendré par l'invasion matériel.

7.2 Les attaques passives

Attaquer un système de sécurité ne se résout pas seulement à attaquer le système en lui même, son implémentation peut être attaquée, c'est ce que l'on

appelle les attaques par "canaux auxiliaires". Ainsi une attaque par canaux auxiliaires ne remet pas en cause la résistance du crypto-système utilisé mais son implémentation, aussi appelées "attaques passives", ce type d'attaque ne nécessite aucune interaction avec le matériel attaqué, ces attaques sont basées sur l'observation des différents signaux : source énergétique, temps d'exécution, etc...

La première publication sur les attaques passives est attribuée à Paul Carl Kocher lors de la conférence CRYPTO qui se déroula à Santa Barbara en 1996. Le canal auxiliaire exploité était la mesure du temps d'exécution du calcul. En mesurant le temps nécessaire pour réaliser une exponentiation modulaire md modulo n pour différents messages, l'attaquant peut bit après bit retrouver la totalité des bits de l'exposant secret d.

En 1998 et en 1999, Paul Carl Kocher présente deux nouvelles technologies qui vont masquer l'histoire de la sécurité embarquée. Il s'agit de l'analyse de puissance simple (traduction littérale de Simple Power Analysis, SPA), et de l'analyse de puissance différentielle (Differential Power Analysis, DPA) sur le DES. Ces deux attaques ont représenté un tournant dans le monde de la carte à puce et ont changé la manière de développer les algorithmes. En 1999 Thomas S.Messerges présente la DPA sur l'exponentiation RSA ainsi que les attaques DPA de second ordre.

Depuis la première DPA, différentes techniques d'attaques plus efficaces ont vu le jour : l'analyse de puissance par corrélation (Correlation Power Analaysis CPA), l'analyse d'information mutuelle (Mutual Information Analysis, MIA), l'analyse en composantes principales, l'analyse par régression linéaire, l'analyse par collision, l'analyse par dictionnaire de courbes de consommation, etc...

7.3 Les attaque actives

Le but de ce type d'attaque est de perturber l'exécution du calcul que réalise le microcircuit. En effet, en perturbant le circuit lors d'une opération celle-ci donne un résultat erroné et donc un résultat cryptographique faux que nous appellerons résultat fauté. En analysant un ou plusieurs de ces calculs erronés, il est possible de retrouver une partie ou la totalité de la clé secrète manipulée.

Chapitre 8

La Proxmark

8.1 Présentation

La Proxmark 3 est une plate-forme RFID-NFC, dotée d'un processeur ARM, d'un FPGA, qui permet de s'adapter aux intervalles de communication imposés par les différentes fréquences, dotée de deux antennes, elle permet de communiquer en basse fréquence (125kHz, 134kHz) et en haute fréquence (13,56 MHz).Développée sous licence GPL par Jonathan Westhues, elle embarque un système d'exploitation permettant de communiquer avec les différentes normes et types de cartes existantes, de fonctionner en toute autonomie, ou de communiquer avec un ordinateur à l'aide d'un logiciel client via USB. Elle peut donc simuler des transpondeurs ou des stations de base et coûte environ 200 euros.

8.2 Prise en main

L'interface utilisateur pour communiquer avec la Proxmark est l'exécutable "proxmark", auquel on doit donner comme paramètre le fichier spécial correspondant au périphérique.

```
./proxmark3 /dev/tty.usbmodem621
proxmark3>
```

On peut alors tester son fonctionnement, on commence par installer l'antenne HF, on présente une carte MiFare Classic, et on lance la commande qui permet d'interagir grâce à la norme ISO 14443. On y récupère alors les informations présentes dans le bloc 0, qui nous permet de connaître le numéro de série de la carte, ainsi que sa version.

```
./proxmark3 /dev/tty.usbmodem621
proxmark3> hf 14a read
ATQA : 04 00
 UID : 72 a3 18 3d
 SAK : 08 [2]
```

```
TYPE : NXP MIFARE CLASSIC 1k | Plus 2k SL1
```

8.3 Communication avec la MiFare Classic

8.3.1 Lecture

Pour lire un bloc on utilise la commande suivante :

```
./proxmark3 /dev/tty.usbmodem621
proxmark3> hf mf rdbl X Y ffffffffffff
```

avec X le numéro de bloc, Y la clef sélectionné (A ou B), et ffffffffff la clef.

8.3.2 Ecriture

Pour écrire des données dans un bloc on utilise la commande suivante :

```
./proxmark3 /dev/tty.usbmodem621
proxmark3> hf mf wrbl X Y ffffffffffff 000102030405060708090
    a0b0c0d0e0f
```

avec X le numéro de bloc, Y la clef sélectionné (A ou B), ffffffffff la clef, et 000102030405060708090a0b0c0d0e0f les données.

Chapitre 9

La MiFare Classic

9.1 Présentation de la MiFare Classic

La MiFare Classic a été créée en 1994, elle a depuis évolué et a connu un grand succès dans le monde. Fin 2010, plus de 3,5 milliards de puces MiFare ont été vendues, avec plus de 40 millions de stations de base. C'est une carte passive, fonctionnant à une fréquence de 13.56 MHz, avec un temps de transaction inférieur à 100 ms, et un débit de 106kbit/s.

9.2 Architecture de la MiFare Classic

9.2.1 Interface

9.3 Structure de la mémoire

Composée d'une mémoire de type EEPROM (Electrically-erasable programmable read-only memory) de 1 kilo octets, elle est découpée en 16 secteurs de 4 blocs, chaque bloc étant constitué de 16 octets. On peut la comparer à une mémoire, ainsi les actions possibles sur la carte sont de simple lecture ou écriture des données contenues dans celle-ci. On a ainsi 768 octets de mémoire utile.

9.3.1 Structure du bloc 0x00

Le premier secteur contient seulement deux blocs de données, un bloc pour l'accès au secteur, et un bloc contentant l'uid de la carte et des informations écrites par le fondeur.

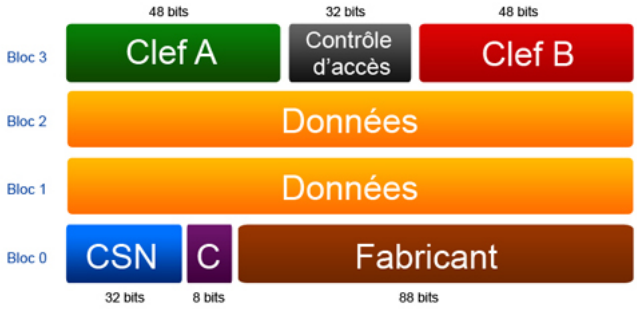

FIGURE 9.1 – Schéma du secteur 0

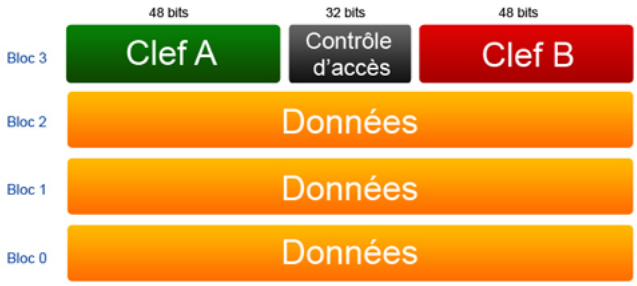

FIGURE 9.2 – Schéma d'un secteur

9.3.2 Structure d'un bloc

Les autres secteurs contiennent trois blocs de données, un bloc pour l'accès au secteur.

9.3.3 Contrôle d'accès à un bloc

Les blocs d'accès aux secteurs permettent de configurer l'accès au secteur, une mauvaise manipulation peut causer des dommages irréversibles au secteur visé, ainsi ces bits permettent de configurer quel clé (A et B) a le droit d'effectuer quelle opération sur le secteur (lecture, écriture).

Chapitre 10

Attaque de la MiFare Classic

10.1 La réto-concenption

La rétro-conception de puces électroniques est parfois utilisée afin de découvrir la logique cablée du circuit, ça a par exemple été le cas pour l'attaque de la MiFare Classic, qui a d'abord subit une rétro-conception par Karsten Nohl, Henryk Plötz, Starbug et David Evans en juillet 2008.

Pour observer des structures élémentaires de plus en plus petites (technologie semi-conducteur à lithographie inférieur à 20 nm) il faut disposer d'outils d'acquisition d'images de grande résolution, comme des microscopes électroniques à balayage. L'abstraction fonctionnelle de ces observations s'appuie sur des équipements informatiques de très grande capacité pour le traitement et l'analyse de volumes d'images colossaux.

Pour passer d'une implémentation physique à la compréhension d'un schéma électronique hiérarchisé,les travaux de rétro-conception s'articulent autour de quatre grandes phases longues et complexes : la préparation des échantillons, la capture et la véctorisation des images, l'identification et le routage des fonctions électroniques de bas niveau et enfin, l'abstraction et la simulation de blocs fonctionnels de haut niveau.

Chaque motif élémentaire fait ensuite l'objet d'une rétro-analyse fonctionnelle à partir des images aux niveau polysilicium et metal1, afin de reconstruire la bibliothèque de primitives électronique de base du fondeur.

À partir de cette représentation informatique du composant, on extrait une "netlist" simulable (VHDL, VERILOG, ...) On procède donc selon une approche Bottom/Up.

Cette analyse électronique fonctionnelle, appuyé par des simulations partielles ou globales du design, permet d'aboutir à une compréhension globale du fonctionnement de la puce.

10.2 CRYPOT1

10.2.1 Présentation

Crypto-1 est un système de chiffrement par flot, développé en 1998 par NXP pour les cartes MiFare Classic, et cassé en 2009 grâce à une cryptanalyse de Garcia, Flavio D. ; Peter van Rossum ; Roel Verdult ; et Ronny Wichers Schreur dans un article intitulé "Wirelessly Pickpocketing a Mifare Classic Card".
Ainsi le but de Crypto-1 est de rendre inintelligible un message codé pour quelqu'un qui ne connait pas la clé, c'est un algorithme à clé secrète, le transpondeur et la station de base possèdent donc cette clé.
Chaque secteur contient un bloc de sécurité (sector trailer), qui stocke deux clefs de 48 bits nommées A et B, ainsi que 4 octets de conditions d'accès au secteur qui sont programmables. La clef A est utilisée pour la lecture, et la clef B pour l'écriture.

10.2.2 Fonctionnement

Posons donc $\mathbb{F}_2 = \{0, 1\}$ représentant les classes du corps $\mathbb{Z}/2\mathbb{Z}$. L'opération d'addition sur ce corps est noté $\oplus$, et représente l'opération "ou exclusif" (XOR). L'opération $\wedge$ représente l'opération "et logique", et $\vee$ représente l'opération "ou logique".

Après la phase d'authentification, la communication entre la station de base et le transpondeur est chiffré.

CRYTO1 est basé sur un LFSR de 48 bits ayant pour polynôme de rétroaction :

$x^{48} + x^{43} + x^{39} + x^{38} + x^{36} + x^{34} + x^{33} + x^{31} + x^{29} + x^{24} + x^{23} + x^{21} + x^{19} + x^{13} + x^9 + x^7 + x^6 + x^5 + 1$

et une fonction booléenne afin de casser cette linéarité :

$f(x_0x_1...x_47) = f_c(f_a(x_9, x_{11}, x_{13}, x_{15}), f_b(x_{17}, x_{19}, x_{21}, x_{23}), f_b(x_{25}, x_{27}, x_{29}, x_{31}), f_a(x_{33}, x_{35}, x_{37}, x_{39}), f_b(x_{41}, x_{43}, x_{45}, x_{47}))$
Avec :
$f_a : \mathbb{F}_2^4 \mapsto \mathbb{F}_2$

$f_a(y_0, y_1, y_2, y_3) = ((y_0 \vee y_1) \oplus (y_0 \wedge y_3) \oplus (y_2 \wedge ((y_0 \oplus y_1) \vee y_3))$

$f_b : \mathbb{F}_2^4 \mapsto \mathbb{F}_2$
$f_b(y_0, y_1, y_2, y_3) = ((y_0 \wedge y_1) \vee y_2) \oplus ((y_0 \oplus y_1) \wedge (y_2 \vee y_3))$

$f_c : \mathbb{F}_2^5 \mapsto \mathbb{F}_2$
$f_c(y_0, y_1, y_2, y_3, y_4) = (y_0 \vee ((y_1 \vee y_4) \wedge (y_3 \oplus y_4))) \oplus ((y_0 \oplus (y_1 \wedge y_3)) \wedge ((y_2 \oplus y_3) \vee (y_1 \wedge y_4)))$

La génération du challenge utilisé pendant l'authentification est produit par un LFSR 16 bits ayant pour polynôme générateur :
$x^{16} + x^{14} + x^{13} + x^{11} + 1$

Sa fonction de rétroaction linéaire est définie par :
$L_{16} : \mathbb{F}_2^{16} \mapsto \mathbb{F}_2$
$L_{16}(x_0 x_1 ... x_1 5) = x_0 \oplus x_2 \oplus x_3 \oplus x_5$

On définit la fonction représentant l'état suivant du LFSR par :
$suc : \mathbb{F}_2^{32} \mapsto \mathbb{F}_2^{32}$
$suc(x_0 x_1 ... x_3 1) = x_1 x_2 ... x_3 1 L_{16}(x_{16} x_{17} ... x_{31})$

Etant donnée que ce générateur pseudo-aléatoire est de 16 bits, sa période est donc de $2^{16} - 1 = 65535$ itérations, et étant donnée qu'une itération se produit toute les $9.44\mu s$, on a alors un cycle complet en 618ms.

Ce générateur se met en route dès que le transpondeur entre dans le champs électromagnétique de la station de base.

10.2.3 Phase d'authentification

La phase d'authentification implémentée est une authentification forte, ainsi lorsque la station de base veut accéder à un bloc et envoie donc une requête, le transpondeur envoie à la station de base un challenge (n_T) A partir de ce moment là, la communication est chiffrée grâce à la clé, l'UID, et le challenge (n_T), on appellera cette key stream k_{s1}. La station de base génère à son tour un challenge (n_R) et répond au transpondeur en lui envoyant la réponse attendue qui est $suc^2(n_T)$ concaténé au challenge généré. La phase d'authentification se termine alors avec l'envoi de $suc^3(n_T)$ du transpondeur vers la station de base.

10.2.4 Les vulnérabilités

La première erreur est au niveau de la taille des clefs, qui font seulement 48 bits.

La deuxième erreur est lié à la génération des nombres pseudo-aléatoire, qui dépendent du temps à partir duquel le transpondeur entre dans le champs élec-

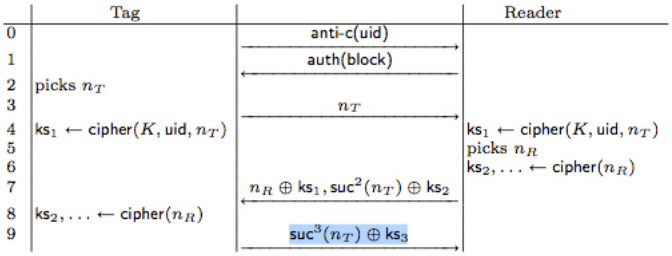

FIGURE 10.1 – Phase d'authentification

tromagnétique, ainsi un attaquant qui maitrise la station de base peut influencer la génération du challenge généré par le transpondeur.

Une troisième erreur de conception est qu'il est possible à cause d'un mauvais choix de la fonction de filtrage du LFSR, permet de revenir à l'état initial du LFSR, qui correspond à la clef.

Une quatrième erreur est qu'une fois l'authentification réussite pour un bloc, il est facile de retrouver facilement la clé d'un autre bloc auquel on tente de s'authentifier, car l'état interne du LFSR change peu.

Une cinquième erreur est que les bits de parités sont calculés sur le texte clair, et non sur le chiffré, ce qui fournit des informations sur la clé.

Chapitre 11

Implémentation d'un contrôle d'accès

11.1 Contexte

Afin de mettre en évidence les problèmes liés à une mauvaise implémentation d'un crypto-système et ses conséquences, nous allons nous mettre dans un contexte d'utilisation nécessitant une forte sécurité : un contrôle d'accès. Celui ci sera développé avec un Seeeduino Mega possédant un processeur ATmega2560, un NFC Shield V2.0 de Seeedstudio basé sur le circuit intégré NXP PN532, et un écran LCD Sparkfun ADM1602K.

11.2 NFC Shield V2.0

Le shield NFC est basé sur le circuit intégré NXP PN532, étant une technologie propriétaire et anciennement secrète, tout dispositif utilisant les MiFare Classic doit posséder un circuit intégré fabriqué par NXP. Ce circuit intégré ne propose qu'une simple API qui permet d'effectuer les opérations de base suivantes :

— Détecter un transpondeur qui vient d'entrer dans le champs
— S'authentifier à un bloc
— Envoyer des trames sur le bloc si l'authentification a réussie
— Lire des trames sur le bloc si l'authentification a réussie
— Configurer le circuit

11.3 Implémentation

Une fois le dispositif connecté on va le programmer afin de lire sur le bloc deux du secteur deux un identifiant qui permettra d'identifier la personne qui

tente de s'introduire dans le bâtiment, si cette personne a alors le droit d'accéder au bâtiment on active un GPIO à l'état logique "1" qui représentera l'ouverture de la porte, sinon le GPIO restera à l'état logique "0" (porte fermé).

On aura donc un projet contenant les fichiers et bibliothèques suivants :

— access.c : Gestion de l'application d'accès.
— db.c : Base de données qui contiendra l'association entre un identifiant et le nom de la personne.
— lcd.c : Gestion de l'affichage LCD.
— La bibliothèque PN532 pour Arduino.
— La bibliothèque LiquidCrystal pour Arduino.

On disposera d'une liaison série entre l'Arduino et l'ordinateur afin d'imprimer les différentes opérations réalisées par le dispositif, celui ci permettra ainsi d'avoir une trace des communications entre celui-ci et la carte MiFare Classic.

11.4 Code source du projet

Le code source du projet est disponible à cette URL : `http://mounirnasrallah.com/cryptoondes`

11.5 Ecriture sur le carte MiFare Classic

Nous avons un deuxième code permettant de configurer une MiFare Classic, en donnant les valeurs de l'ancienne clé, de la nouvelle clé, du numéro du secteur, du numéro du bloc, et de des données à envoyer, on peut alors écrire des données ou modifier la clé.

11.6 Fonctionnement

On obtient le dispositif suivant :

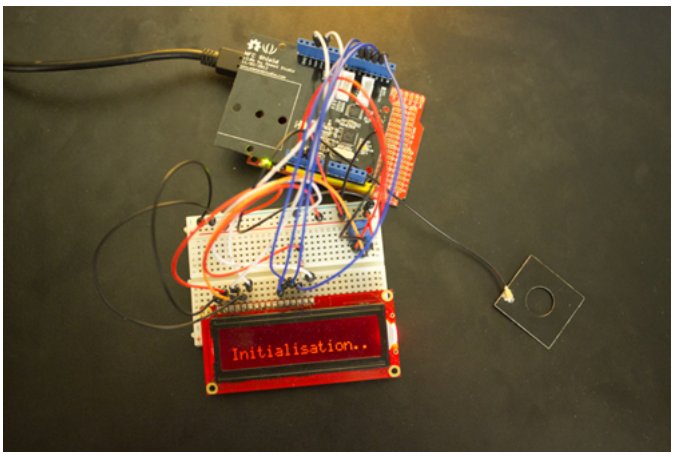

FIGURE 11.1 – Initialisation du système

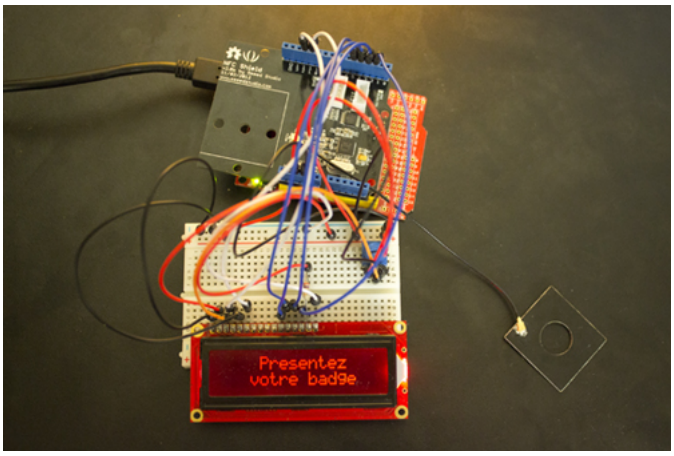

FIGURE 11.2 – Attente d'un transpondeur

Chapitre 12

Attaque d'une MiFare Classic

On va maintenant attaquer une carte MiFare Classic en utilisant les vulnérabilités du crypto-système CRYPTO1, ainsi la Proxmark permet d'utiliser toutes ces vulnérabilités. Ainsi le fichier crapto1.c contient l'algorithme pour attaquer efficacement une carte en une dizaine de secondes. Cet algorithme utilise la vulnérabilité des bits de parité, et la linéarité du crypto-système afin de retrouver l'état initial du LFSR.

```
proxmark3> hf mf mifare
--------------------------------------------------------------------------

Executing command. Expected execution time: 25sec on average  :-)
Press the key on the proxmark3 device to abort both proxmark3 and
    client.
--------------------------------------------------------------------------

.......................


Can't open logfile, logging disabled!

uid(72a3183d) nt(5cd6b2af) par(5da5d5fda57dcd45) ks(0
    c00080b040c0d09) nr(00000000)


|diff|{nr}    |ks3|ks3^5|parity         |
+----+--------+---+-----+---------------+
| 00 |00000000| c |  9  |1,0,1,1,1,0,1,0|
| 20 |00000020| 0 |  5  |1,0,1,0,0,1,0,1|
| 40 |00000040| 8 |  d  |1,0,1,0,1,0,1,1|
| 60 |00000060| b |  e  |1,0,1,1,1,1,1,1|
| 80 |00000080| 4 |  1  |1,0,1,0,0,1,0,1|
| a0 |000000a0| c |  9  |1,0,1,1,1,1,1,0|
| c0 |000000c0| d |  8  |1,0,1,1,0,0,1,1|
| e0 |000000e0| 9 |  c  |1,0,1,0,0,0,1,0|
key_count:1
```

```
25 ----------------------------------------------------------------
26 Key found:8829da9daf76
27
28 Found valid key:8829da9daf76
```

Chapitre 13

Et la vie privée ?

La technologie RFID soulève de nombreuses questions quant à la vie privée, étant donné que l'utilisateur n'a aucune action physique à effectuer pour activer le transpondeur, il suffit souvent que le transpondeur soit dans le champ électromagnétique de la station de base pour qu'elle puisse l'activer et le faire "parler". Ce qui pose un problème de vie privée, puisque cette technologie permet donc de traquer les personnes dans leur quotidien (itinéraire dans les transports en commun,traçabilité dans les lieux publics ,récupération des dernières transactions bancaires, ...).

Il suffit de regarder le brevet d'IBM intitulé " Identification and Tracking of Persons Using RFID Tagged Objects". La stratégie d'IBM est de dissimuler des lecteurs RFID appelés "person tracking units" dans des lieux publics comme des commerces, des lieux de manifestations sportives, des cinémas, des bibliothèques, des musées ... Ils permettraient aux personnes chargées du marketing et aux forces de l'ordre de surveiller les individus à partir des signaux émis par les transpondeurs des objets qu'ils portent. En faisant le lien entre les fichiers des commerçants qui identifient les clients qui achètent des articles et la lecture des transpondeurs de ces articles.

D'autres compagnies, comme Philips, Procter & Gamble et Intel, aimeraient que la RFID soit étendue à nos foyers ou ils pourraient garder un œil sur le contenu de nos réfrigérateurs ou de nos armoires à pharmacie.

Les services de police qui voudraient identifier les participants à une manifestation, pourraient le faire simplement en s'y infiltrant avec des stations de base portables cachés dans leurs sacs à dos.

Chapitre 14

Pour aller plus loin

14.1 La bibliothèque François Mitterand

La carte d'abonnement de la bibliothèque François Mitterand utilise des MiFare Classic, ce qui malheureusement constitue une grave erreur depuis la mise à nue du crypto-système CRYPTO1, ainsi il est possible de dupliquer des cartes d'étudiants présent dans la salles de lecture à leur insu grâce à une Proxmark III et une antenne permettant de communiquer à quelques mètres, ce qui est facilement réalisable. De plus la clef utilisée fait partie des clefs par défaut, et possède donc des suites de symboles prévisibles.

14.2 La Carte Navigo

La carte Navigo repose elle sur un tout autre standard, nommée "Calypso". Ce standard utilise la norme appelée 14443 Type B', trop souvent appelé à tort ISO 14443 B' mais n'existe pas à l'ISO, et un standard pour le stockage des données et des différentes applications que peut embarquer la carte.

Elle est toute fois lisible grâce à un simple lecteur de carte à puce, vendue par la SNCF et la RATP pour la modique somme de sept euros. Initialement ce lecteur permet de recharger son Pass Navigo de chez soi via une applet, mais peut être utilisé pour lire ou écrire des données sur toutes les cartes à puces en installant simplement les pilotes pour le système d'exploitation de sa machine.

Après avoir téléchargé les sources du logiciel cardpeek à l'adresse suivante : `https://code.google.com/p/cardpeek/`, on compile les sources puis on lance l'application via la commande suivante :

```
1 ./cardpeek
```

On obtient alors une interface graphique qui nous permet de sélectionner le lecteur de carte à puce que l'on veut utiliser, et la norme de la carte présente pour

FIGURE 14.1 – Lecture d'une carte Navigo

tenter de la lire. On lit alors toutes les informations contenues dans la carte : les dernières utilisations de la carte avec la station dans laquelle on a validé notre titre, la date et l'heure, le type de transport utilisé, le forfait présent dans la carte, l'historique des abonnements qui ont été pris, et d'autres informations.

14.3 Les passeports biométriques

L'utilisation d'un tag RFID dans les passeports fit son apparition en Malaisie en 1998, et c'est dans les passeports belges que l'on trouve les premiers tags qui répondent au standard de l'Organisation de l'Aviation Civile Internationale (ICAO). La France a délivré quant à elle, ses premiers passeports biométriques en avril 2006.

L'utilisation d'un dispositif électronique dans les passeports a pour objectif de renforcer leur sécurité, en intégrant un tag RFID dans le passeport, celui-ci devient en effet inviolable, en effet, il devient théoriquement impossible pour un faussaire de modifier son contenu, il est également impossible d'en créer un nouveau ou d'en dupliquer un.

Les passeports utilisent des tags performants dont le prix peut coûter plusieurs euros. Ils sont munis d'un microprocesseur capable de réaliser des opérations cryptographiques symétrique et asymétrique, ce qui demande pour le dernier cas une capacité de calcul élevée. Ils possèdent également une grande quantité de mémoire afin de stocker les données biométriques. Le standard de l'ICAO impose également que le transpondeur soit compatible avec le standard ISO 14443 (type A ou B).

Selon le standard de l'ICAO, le transpondeur doit au moins contenir les informations de bases suivantes : le nom du porteur, prénoms, date de naissance,

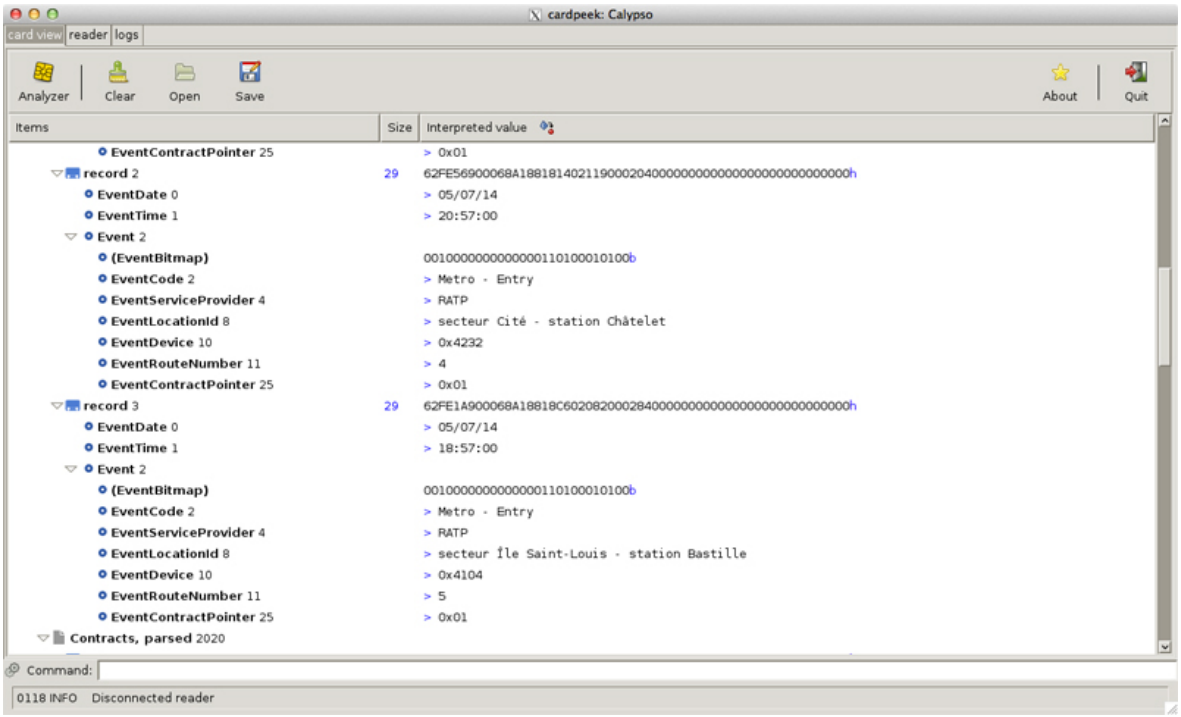

FIGURE 14.2 – Informations personnels sur les trajets enregistrés sur la carte Navigo

FIGURE 14.3 – MRZ présent dans un passeport français.

sexe, date d'expiration, nationalité, autorité ayant émis le passeport, ainsi qu'une photo numérique du porteur (format JPEG, environ 20 Ko). Le standard prévoit également que le tag puisse contenir des informations optionnelles comme la signature manuscrite du porteur, son adresse, son numéro de téléphone, ...

Il est possible de lire les données présentes dans les passeports grâce au projet RFIDIOt, et d'une simple station de base. Il est par contre nécessaire de générer une clef grâce aux symboles figurant en bas de la page contenant les informations personnelles, on appelle cette zone : MRZ (machine-readable zone). Elle est ainsi lu par ordinateur via un simple dispositif de capture vidéo, ce qui oblige un attaquant d'avoir une vue sur cette page.

Chapitre 15

Références

1 - Applications en identification radiofréquence et cartes à puce sans contact, Dominique Paret, Dunod Paris.

2 - RFID handbook : fundamentals and applications in contactless smart cards and identification / Klaus Finkenzeller, traduit par Rachel Waddington, J. Wiley & sons, Chichester

3 - NFC, Near Field Communication : principes et applications de la communication en champ propre, Dominique Paret, Xavier Boutonnier, Youssef Houiti, Dunod, Paris

4 - RFID security : techniques, protocols and system-on-chip design, Yan Zhang, Springer, Paris Kitsos

5 - A Practical Attack on the MIFARE Classic, Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D. Garcia, Institute for Computing and Information Sciences, Radboud University Nijmegen

6 - AN10834 MIFARE ISO/IEC 14443 PICC Selection, Rev. 3.0 — 26 June 2009, NXP

7 - Cryptanalytic Attacks on MIFARE Classic Protocol, Jovan Golic, Security Lab, Telecom Italia IT, RSA Conférence 2013

8 - Analysis of the Algebraic Side Channel Attack, Claude Carlet, Jean-Charles Faugère, Christopher Goyet, Guénaël Renault, published in "Journal of Cryptographic Engineering 2, 1 (2012) 45-62"

9 - Applying Remote Side - Channel Analysis Attacks on a Security - enabled NFC Tag, Thomas Korak, Thomas Plos, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Austria,

10 - MF1S70yyX MIFARE Classic 4K - Mainstream contactless smart card IC for fast and easy solution development, Rev. 3.0 — 2 May 2011, NXP

11 - Tutorial : Proxmark, the Swiss Army Knife for RFID Security Research, Flavio D. Garcia, Gerhard de Koning Gans, and Roel Verdult, Institute for Computing and Information Sciences, Digital Security Group, Radboud University Nijmegen, The Netherlands.

12 - Dismantling MIFARE Classic, Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs, Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands

13 - Cryptographie appliquée : algorithmes protocoles et codes source en C, Bruce Schneier traduction de Laurent Viennot, 2ème edition, Vuibert, Paris

14 - "La cryptographie militaire" du Journal des sciences militaires (vol. IX, Janvier 1883, Février 1883)