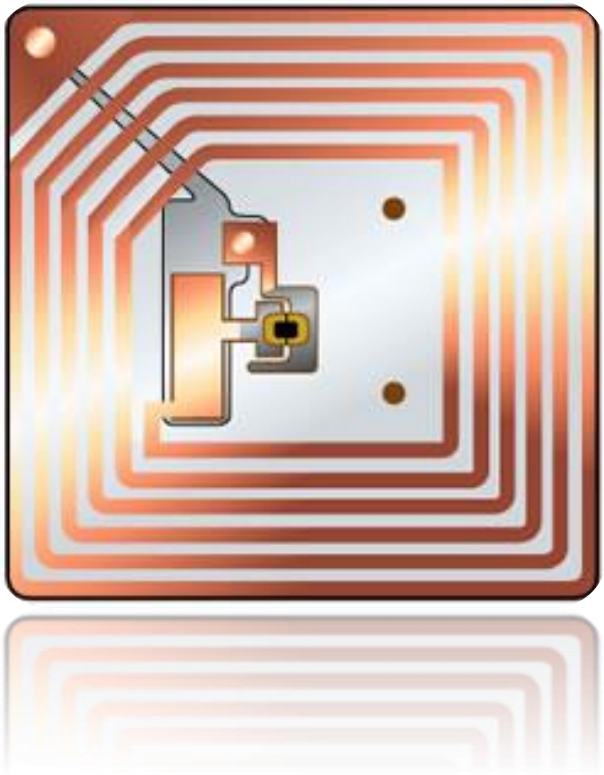
SY22 P14



SYSTEMES SANS FIL

RFID

Table des matières

Introduction

L'UV SY22 nous permet d'étudier de façon assez complète les technologies sans fil (Wifi, Bluetooth, etc.). Ainsi, à l'issue de cette étude, nous devions choisir une technologie sans fil que nous avions étudié lors des cours pour pousser l'étude un peu plus loin. Ici, nous avons choisi le RFID.

Le RFID est une technologie très répandue et extrêmement discrète, ce qui lui permet d'être présente partout. C'est une communication sans fil de courte distance (quelques centimètres en général) entre une station et un récepteur passif (non alimenté). Ces récepteurs sont disponibles en de nombreuses tailles, les plus gros étant au format carte bleu et les plus petits sont presque de même taille qu'un grain de riz !

Cette technologie est utilisée dans de nombreux domaines très étendus : paiement, antivol, marquage d'animaux, chargement, etc. Cependant, son usage principal reste le contrôle d'accès via l'identification d'un récepteur en particulier.

Projet

Notre projet est d'étudier la technologie RFID et particulièrement un certain type de puces : les puces MIFARE Classic, conçues par NXP. Ces puces sont les plus répandues, elles représentent 85% du marché RFID. Nous pouvons notamment les retrouver dans les cartes UTT.

Dans ce dossier, nous allons tout d'abord étudier la communication entre la station et la carte, puis les informations présentes dans cette dernière.

Pour cela, il nous a été fourni un set de 7 cartes MIFARE 1k vierges et un kit de développement RFID Netronix MD-003 contenant une station antenne et un microcontrôleur MMU57D :

Ce microcontrôleur doit être relié au PC via un port série et la communication est assurée par un logiciel fonctionnant sous Windows. Cependant, il est possible de communiquer directement avec la station via un terminal série, le protocole étant bien documenté.

L'idée du projet est d'exploiter cette station pour récupérer les informations présentes sur une carte MIFARE et tenter de réaliser une copie de cette dernière.

La technologie RFID

Communication

En introduction, il est précisé que seule la station est pourvue d'énergie. En effet, le récepteur doit s'alimenter dessus pour fonctionner. Pour cela, l'induction est utilisée : l'énergie nécessaire au fonctionnement du récepteur est transmise sans fil grâce à une bobine dans laquelle le courant circule. Cela provoque un champ électrique, qui est capté par le récepteur via une autre bobine, cette dernière étant reliée à l'alimentation de la puce du récepteur.

Maintenant que le récepteur est alimenté, il va pouvoir communiquer. Pour cela, il va utiliser les mêmes bobines : le récepteur fait varier la charge qu'il applique sur sa bobine pour transmettre des informations, ce qui va faire varier la charge sur l'autre bobine présente sur la station.

Pour le reste du projet, nous nous sommes uniquement consacrés à des cartes MIFARE 1K.

Stockage des informations sur la carte

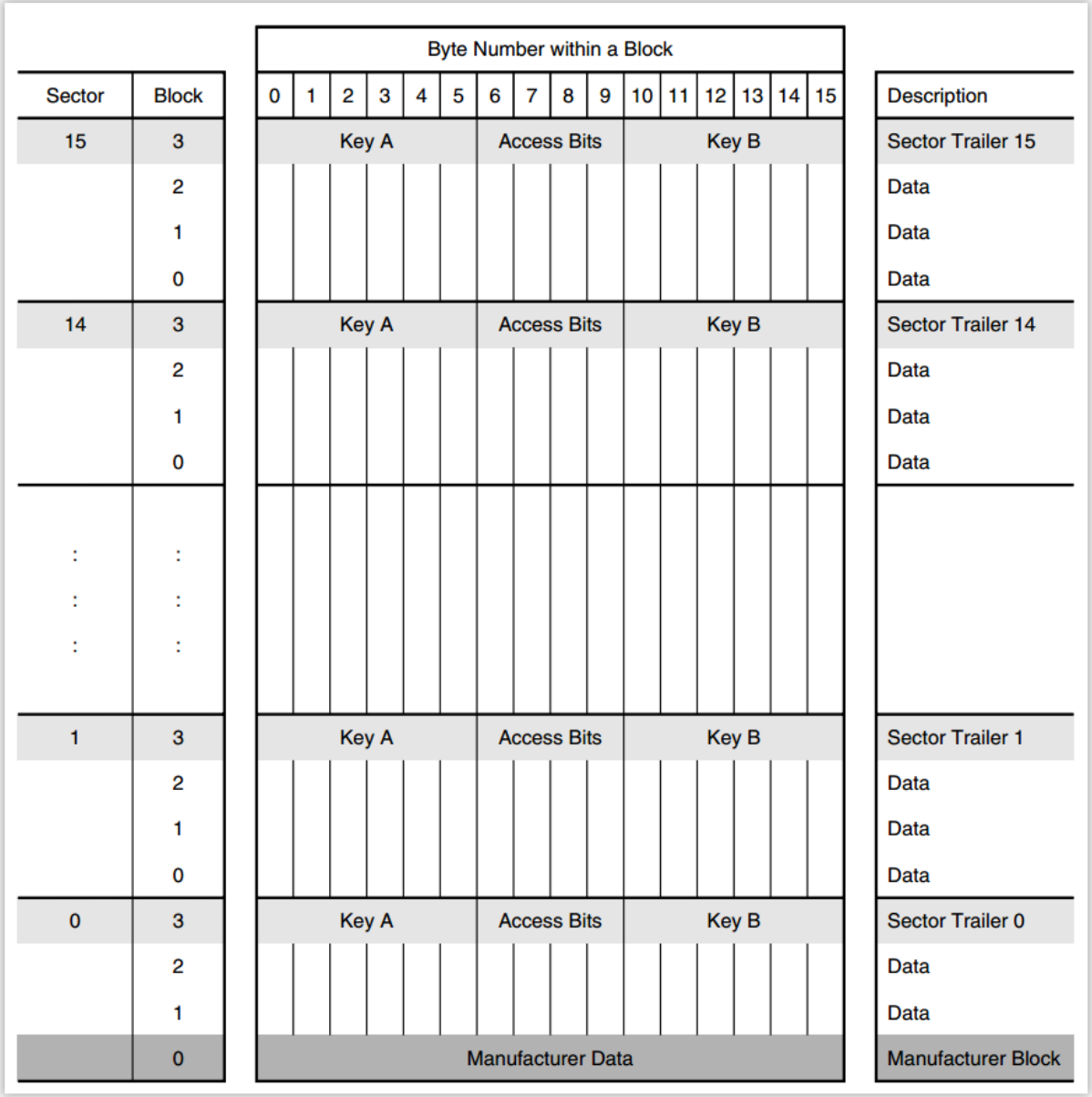
La carte MIFARE 1k dispose de 1024 octets de stockage. La mémoire est divisée en 16 secteurs (de 0 à 15) et chacun de ces secteurs est séparé en 4 blocs (numérotés de façon absolue de 0 à 63), qui peut contenir 16 octets chacun.

Le dernier bloc d'un secteur sert à définir les conditions d'accès à chacun des blocs de ce secteur. Il contient une clé d'accès A de 6 octets, les conditions d'accès sur 3 octets, un octet inutilisé et une clé B de 6 octets. Ainsi, chaque secteur peut avoir ses propres conditions d'accès indépendamment des autres.

Le premier bloc du premier secteur (bloc 0) est spécial dans la mesure où il est en lecture seule. En effet, il contient les informations liées au fabriquant de la carte : numéro de série sur 7 octets et le reste du bloc, qui n'est pas normé et laissé pour le constructeur.

Remarque : il est possible de trouver sur internet des cartes sur lesquelles le bloc 0 est réinscriptible, il est donc envisageable de copier parfaitement une carte MIFARE.

Ce schéma illustre l'organisation de la mémoire de la carte :



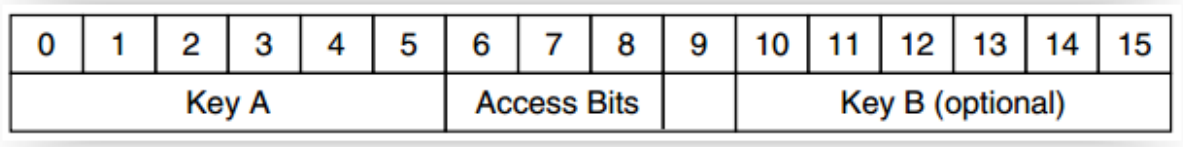
Sector	Block	Byte Number within a Block 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
15	3	Key A						Access Bits				Key B						Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data
14	3	Key A						Access Bits				Key B						Sector Trailer 14
	2																	Data
	1																	Data
	0																	Data
:	:																	
:	:																	
:	:																	
1	3	Key A						Access Bits				Key B						Sector Trailer 1
	2																	Data
	1																	Data
	0																	Data
0	3	Key A						Access Bits				Key B						Sector Trailer 0
	2																	Data
	1																	Data
	0	Manufacturer Data																Manufacturer Block

Remarque : la charge utile de chaque carte est donc égale à 768 octets.

Les ACL (Access Conditions)

Les conditions d'accès sont stockées dans le dernier secteur de chaque bloc. Ils permettent de définir sous quelles conditions il est possible de lire ou d'écrire sur le secteur de la carte. Ces Access bits sont à manipuler avec précautions puisqu'une mauvaise information pourrait corrompre le secteur et le bloquer en l'état. L'algorithme CRYPTO1 est ainsi implémenté, il s'agit d'un algorithme d'encryptions créé par NXP Semiconductors pour les tags MIFARE.

Voici en détail les informations contenues sur ce bloc :



Remarque : l'octet 9 est rarement utilisé, c'est un octet optionnel.

La clé A

La clé A est obligatoire pour lire un secteur. Elle n'est évidemment pas lisible puisqu'il s'agit de la sécurité de base sur les cartes MIFARE.

Par défaut, cette clé est FF FF FF FF FF FF. Il existe aussi une dizaine de clés « standard » avec lesquelles une lecture pourrait être tentée :

A0 B0 C0 D0 E0 F0
A1 B1 C1 D1 E1 F1
A0 A1 A2 A3 A4 A5
B0 B1 B2 B3 B4 B5
4D 3A 99 C3 51 DD
1A 98 2C 7E 45 9A
00 00 00 00 00 00
D3 F7 D3 F7 D3 F7
AA BB CC DD EE FF

Cette clé permet l'accès à certaines informations, définies par les Access bits.

Access bits

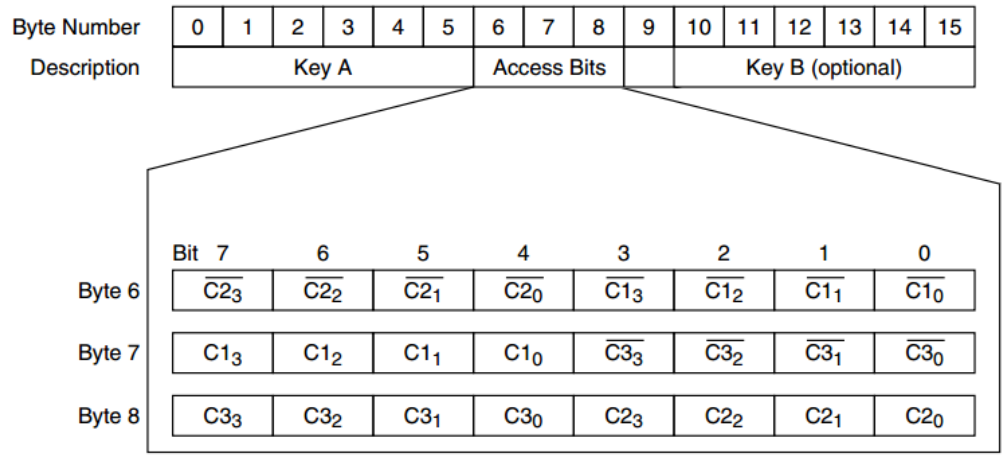
Les Access bits peuvent permettre ou interdire certaines opérations sur les blocs.

Voici les différents types d'opérations disponibles :

- Lecture
- Ecriture
- Incrémentation
- Décrémentation
- Transfert (déplacement)
- Restauration

Seules les deux premières modifications (lecture et écriture) sont possibles sur le bloc contenant les Access bits. Les informations des 3 octets d'Access bits sont redondées pour assurer leur exactitude.

Les Access bits sont organisés de la façon suivante :



Soit **CXy** le bit contenant l'information d'accès **X** du bloc **y** du secteur

Les informations d'accès sont codées de la façon suivante :

Access bits			Conditions d'accès pour :						Remarque
			Clé A		Access bits		Clé B		
C1	C2	C3	Lecture	Ecriture	Lecture	Ecriture	Lecture	Ecriture	
0	0	0	Jamais	Clé A	Clé A	Jamais	Clé A	Clé A	La clé B peut être lue
0	1	0	Jamais	Jamais	Clé A	Jamais	Clé A	Jamais	La clé 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	La clé 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	

Remarque : nous pouvons observer que la clé A n'est jamais lisible et que, dans certaines conditions, il est possible de bloquer un bloc en lecture seule.

Authentification avec une carte MIFARE

Plusieurs techniques peuvent être mises en œuvre pour authentifier un utilisateur via sa carte :

- la première est de ne même pas stocker d'informations sur la carte et d'utiliser simplement le numéro de série de la carte (présent dans le bloc 0 et transmis lors du premier contact avec la station).

- la deuxième consiste à lire un identifiant stocké en mémoire, en clair ou chiffré avec une clé privée présente sur le terminal d'authentification.

- la dernière consiste à stocker des données sur la carte : les abonnements et l'historique des passages, il ne s'agit donc pas que d'un simple identifiant. Cette utilisation est la moins sécurisée et donc la moins fréquente.

Nous prouverons plus loin qu'aucune de ces solutions n'est fiable et que toutes les sociétés qui utilisaient ce mode d'authentification migrent rapidement vers des technologies plus sûres à base de cartes à microcontrôleur.

Travail réalisé

Programme de duplication de carte (en C)

Le programme

Nous avons réalisé un programme en C qui permet de copier/dumper tout le contenu d'une carte et d'écrire ce même dump sur une autre carte. Le programme communique directement les commandes à la puce MMU57D via le port série pour successivement :

1. initialiser la communication avec la carte
2. enregistrer les informations d'authentification en mémoire tampon sur la station
3. s'authentifier à un secteur en utilisant une courte liste des clefs les plus courantes
4. écrire ou lire tous les blocs, avant de passer au secteur suivant (étape 2)

L'application s'utilise comme suit :

Lecture de la carte et écriture de son contenu dans toto.bin : **./nfc -r toto.bin**

Lecture de toto.bin et écriture de son contenu sur la carte : **./nfc -w toto.bin**

Evidemment, si la carte utilise des clefs non triviales sur certains de ses secteurs, nous ne pouvons pas y avoir accès. Cependant, nous avons bien essayé de pénétrer chaque secteur par une attaque par force brute mais la puce met trop de temps à répondre lorsqu'on lui envoie une requête d'authentification (~200 ms). En effet, sachant qu'une clef a une taille de 6 octets (ou 48 bits), il y a 2^48 = 281474976710656 combinaisons différentes possibles. Si le test d'une combinaison met 200 ms à se faire, tester toutes les combinaisons mettrais plusieurs milliers d'années avant d'avoir un résultat.

Il a donc fallu trouver une solution alternative. Cependant, si un secteur n'est pas ouvrable avec une clé standard, nous offrons la possibilité à l'utilisateur d'entrer lui-même un set de clés correctes.

Test avec la carte étudiant

En essayant de copier la carte étudiant, nous avons remarqué que certains secteurs sont disponibles avec des clés standard, et d'autres non.

Voici un dump de la carte :

```
Sector 0 (0x00)
[00] r--  8A 42 91 FD A4 88 04 00 48 85 14 59 61 70 48 10 |.B......H..YapH.|
[01] rW-  A8 00 42 58 42 58 42 58 43 58 43 58 43 58 44 58 |..BXBXBXCXCXCXDX|
[02] rW-  00 00 00 00 00 00 00 00 73 88 00 00 00 00 34 48 |........s.....4H|
[03] WXW  A0:A1:A2:A3:A4:A5 78:77:88 C1 XX:XX:XX:XX:XX:XX
          MAD access key                (unknown key)

Sector 1 (0x01)
[04] rW-  11 01 02 31 33 31 39 36 00 00 00 17 28 19 7E 00 |...13196....(.~.|
[05] rW-  32 32 30 30 30 30 30 30 31 33 31 39 36 35 B7 6D |22000000131965.m|
[06] rW-  31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |1...............|
```

```
[07] WXW  A0:A1:A2:A3:A4:A5 78:77:88 00 XX:XX:XX:XX:XX:XX
          MAD access key                (unknown key)

Sector 2 (0x02)
[08] rW-  21 00 00 00 00 00 00 00 00 00 00 00 00 00 4D |!.............M|
[09] rW-  41 52 54 49 4E 00 00 00 00 00 00 00 00 00 00 |ARTIN...........|
[0A] rW-  00 00 00 00 00 00 00 00 00 00 00 00 00 56 69 63 |.............Vic|
[0B] WXW  A0:A1:A2:A3:A4:A5 78:77:88 00 XX:XX:XX:XX:XX:XX
          MAD access key                (unknown key)

Sector 3 (0x03)
[0C] rW-  31 74 6F 72 00 00 00 00 00 00 00 00 00 00 00 |1tor............|
[0D] rW-  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
[0E] rW-  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
[0F] WXW  A0:A1:A2:A3:A4:A5 78:77:88 00 XX:XX:XX:XX:XX:XX
          MAD access key                (unknown key)

Sector 4 (0x04)
[10] ???  -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[11] ???  -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[12] ???  -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[03] ???  XX:XX:XX:XX:XX:XX --:--:-- -- XX:XX:XX:XX:XX:XX
          (unknown key)                 (unknown key)
[…]
Sector 8 (0x08)
[20] rwi  31 C3 E5 DB 80 23 64 8B 64 A6 54 49 5E 04 50 AE |1....#d.d.TI^.P.|
[21] rwi  D3 9F 35 74 CF D6 14 66 82 A4 9B 73 AD 5F C5 21 |..5t...f...s._.!|
[22] rwi  96 C3 CA 9B 94 AA 80 4C 35 10 F5 98 BB 05 F3 46 |.......L5......F|
[23] WXW  FF:FF:FF:FF:FF:FF 7F:07:88 69 XX:XX:XX:XX:XX:XX
          Factory default key           (unknown key)
```

Les secteurs 4, 5, 6, 7, 12 et 15 utilisent des clés non standard, il n'a donc pas été possible de les ouvrir avec notre programme. Les secteurs 9, 10, 11, 13 et 14 sont vides.

La plupart des secteurs sont encodés avec les Access conditions suivantes : 78 77 88. Cela signifie qu'il n'est possible d'écrire qu'en connaissant la clé B, qui est inconnue sur tous les secteurs.

Il n'est pas possible de faire une parfaite copie de la carte dans ces conditions, il faut absolument réussir à ouvrir tous les secteurs. Pour cela, nous avons tenté une autre approche.

Dump d'une carte sous Kali linux

Introduction

Pour aller plus loin et réussir à exploiter les failles de la technologie MIFARE, il nous a fallu investir dans un autre matériel, à savoir un lecteur NFC USB ACR122U. Ce matériel étant compatible avec la librairie *libNFC* sous linux.

Nous avons également installé une version de Linux dédiée à la sécurité offensive, elle intègre directement tous les outils pour tester les failles MIFARE mais pas seulement…

Nous avons pu tester un programme en C intitulé mfoc qui permet, si on dispose d'au moins une clé, de retrouver toutes les autres. Son implémentation est cependant assez obscure, mais nous sommes surs que le programme n'utilise pas de force brute, il exploite une faille de l'algorithme CRYPTO1 conçu par NXP.

Cet algorithme tenu secret par NXP a été découvert en 2007 par Karsten Nohl et Henryk Plötz, pour retrouver l'algorithme, ils ont observé au microscope la puce MIFARE conçue par NXP et l'ont décortiqué en la découpant en ce qui devait être 10 000 couches, pour identifier toutes les portes logiques de la puce. L'étude s'est avérée plus « simple » car la puce était en fait modélisable en 70 formules matlab… En continuant l'étude, ils ont aussi remarqué que le générateur pseudo-aléatoire de la puce n'était pas si aléatoire, il utilisait le temps depuis lequel la puce est sous tension pour générer les nombres, il a donc été possible de recréer ce générateur. Cependant, ils n'ont pas révélé assez de détails pour permettre la création d'un programme, mais cela a motivé de nombreuses communautés de développeurs, ce qui a fait naitre en 2008 les premiers exploits. Plus de détails sont disponibles dans les documents fournis dans la section Glossaire.

Par la suite de nombreux programmes sont sortis pour permettre l'exploitation de ces failles le plus simplement possible, les deux plus connus étant mfoc et mfcuk, le premier déjà évoqué permet de retrouver les clés d'une carte en en connaissant déjà au moins une, et le deuxième permettant de retrouver une clé sur une carte dont toutes les clés sont inconnues.

Nous avons testé ces deux programmes, le premier avec la carte d'étudiant nous a permis de retrouver toutes les clés en moins de 5 minutes, en nous offrant la possibilité de lire et d'écrire tous les secteurs..

Nous avons testé mfcuk avec une carte dont on ne connaissait aucune clé (badge d'accès immeuble), et mfcuk a pu retrouver la clé à chaque fois en moins d'une demi-heure.

Test avec la carte étudiant

La carte d'étudiant a pu être sauvegardée intégralement, ce qui nous a permis d'analyser plus en détail son contenu. Nos contacts dans le milieu associatif nous ont permis d'économiser un peu de temps, l'analyse en détail de la carte ayant déjà été faite par des collègues. Ils nous ont aidé à identifier certaines failles dont une qui concerne le service impressions…

Pour aller plus loin, nous avons essayé de savoir s'il était possible d'ajouter les accès au bâtiment D sur nos cartes, et il s'est avéré que c'est impossible, en effet, la carte d'étudiant contient une clé générée à partir du numéro de série de la carte et avec une clé privée connue de l'administrateur du système. La même clé privée est contenue dans la serrure électronique. Cette sécurité empêche l'ajout de

l'accès sur une carte mais rien n'empêche de copier une carte contenant les accès, y compris le numéro de série, il est possible de se procurer en ligne des cartes dont le manufacturer block (bloc 0) est totalement inscriptible et donc de faire une copie parfaite de la carte.

D'autres cartes

Les badges d'immeubles ne contenant aucune information, il est très simple de les copier, il suffit simplement de connaitre la clé A, avec mfcuk, les authentifications ne sont pas basées sur le numéro de série, mais simplement sur la clé A, et tous les autres secteurs ont la même clé. il n'est donc pas nécessaire d'utiliser une carte spéciale.

Les cartes de transport de la TCAT sont plus sécurisées car elles ne contiennent aucune information, la carte est simplement identifiée et les informations sur l'abonnement sont récupérées sur un serveur.

Pour toutes les cartes « rechargeables en lignes », il est inutile de les modifier pour obtenir un service, les informations sont stockées sur un serveur. Mais il est possible dans tous les cas de les copier et d'utiliser le même abonnement pour plusieurs personnes.

Pour aller plus loin

La communication NFC est désormais utilisée pour le paiement sans contact, il serait intéressant d'étudier l'implémentation d'un tel système, étudier ses failles savoir si des usages plus obscurs sont possibles avec ces cartes. Il est possible de concevoir des antennes très simplement, capables d'avoir une portée de plusieurs mètres !

Nous nous sommes très rapidement intéressés à l'émulation de carte, mais notre lecteur NFC n'étant pas compatible, nous ne sommes pas allés plus loin.

Il est possible également d'espionner une communication entre un lecteur et un tag NFC, cela avec des outils très simple : une carte son à moins de 10€ et quelques composants, nous avons mis un lien dans le glossaire présentant de la documentation à ce sujet et une distribution Linux dédiée à cet usage : OpenPCD.

Conclusion

Cette étude nous a permis d'en apprendre plus sur une technologie utilisée au quotidien, le NFC, il est important de se pencher sur les problématiques de sécurité liées à l'utilisation des cartes. Les domaines d'applications de cette technologie sont en pleine expansion, le paiement fait partie d'un des nouveaux usages où la sécurité est primordiale et largement perfectible.

Sachant qu'il est possible d'aller plus loin, utiliser un autre type de cartes étudier d'autres usages (paiement, antivol, etc.) une étude sur l'émulation de carte peut aussi avoir un intérêt certain, surtout depuis que les terminaux Android sont équipées de cette technologie.

Cette UV nous a surtout ouvert l'esprit sur le panel de possibilités offertes par le RFID et surtout l'importance de la sécurité sur ce type de communications.

Glossaire

Documentation NXP MIFARE 1k : http://www.nxp.com/documents/data_sheet/MF1S50YYX.pdf

MFCUK : https://code.google.com/p/mfcuk/

Mfoc : https://code.google.com/p/mfoc/

24C3 : Karsten Nohl, Henryk Plötz : https://www.youtube.com/watch?v=QJyxUvMGLr0

Documentation MIFARE weaknesses : http://eprint.iacr.org/2009/137.pdf

Documentation écoute communication : http://www.openpcd.org/Live_RFID_Hacking_System