Ambroise Decouttere et Raphaël Tournafond

Pickpocketing Mifare

Table des matières

Introduction

Avec plus de 1 milliards de carte vendue, la Mifare Classic représente plus de 70% des cartes à puce sur le marché.

La Mifare Classic est notamment très utilisé dans les systèmes de paiement des transports public comme par exemple les cartes Oyster dans le métro de Londres ou encore la Charlie Card à Boston. Elle est également beaucoup utilisée pour le contrôle des accès de bureau ou de bâtiment gouvernementaux.

La Mifare Classic ne respecte pas tous les standards de sécurité. C'est le protocole de transmission qui ne respecte pas les standards puisque la Mifare Classic utilise sa propre couche de communication sécurisée. Pour permettre une confidentialité dans les échanges et une authentification entre la carte et le lecteur, celle-ci utilise le système de chiffrement CRYPTO1.

Nous allons donc voir quelles sont les vulnérabilités de la Mifare Classic qui permettent à un attaquant de récupérer les clés cryptographiques de la carte simplement à l'aide d'une communication sans fil. Car les potentiels attaque peuvent avoir un gros impact puisqu'elles permettent par exemple « voler » la carte en la copiant.

Histoire

La technologie de MIFARE Classic (MIkron FARE-collection System) a été développée par Mikron puis acquise par Philips en 1998. Mikron transmit la licence de la technologie MIFARE Classic à Atmel aux États-Unis, à Philips aux Pays-Bas et à Siemens (maintenant Infineon Technologies) en Allemagne.

Après l'acquisition par Philips, Hitachi a contracté une licence de produits MIFARE avec Philips pour le développement de solutions à base de cartes à puce sans contact pour la téléphonie à carte de NTT entre 1999 et 2006. Trois autres partenaires ont rejoint le projet des cartes sans contact pour les téléphones : Tokin-Tamura-Siemens, Hitachi (en contrat avec Philips pour le support technique), Denso (Motorola pour la production). NTT demanda deux versions de la puce, comme MIFARE Classic, une avec petite capacité mémoire et l'autre avec une grande capacité mémoire. Hitachi a développé seulement la carte à grande capacité et retira une partie de la mémoire pour la version à faible capacité.

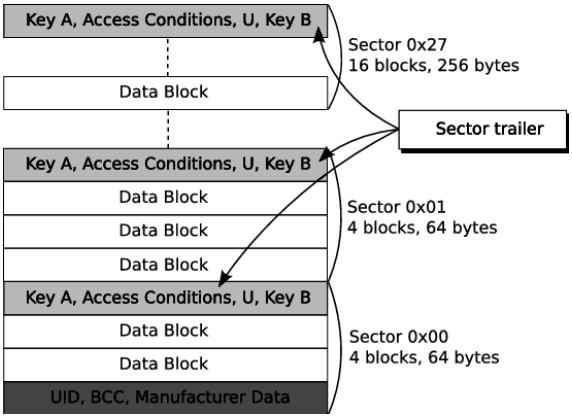
Siemens a développé seulement la puce à logique câblée basée sur leur technologie MIFARE avec quelques modifications. Motorola tenta de développer une puce à logique câblée similaire à MIFARE mais finit par abandonner. Le projet visait la production d'un million de cartes par mois pour commencer mais atteint 100 000 cartes par mois juste avant qu'il soit abandonné.

La technologie

Description

La carte Mifare Classic est une puce mémoire avec un système de communication sans fil sécurisé. La mémoire de la puce est partagée en plusieurs parties, chacune d'entre elle est divisée en blocs de 16 octets chacune. Le dernier bloc de chaque partie contient 2 clés secrète et les conditions d'accès pour la partie en question.

Pour effectuer une opération sur une bloc spécifique, le lecteur doit d'abord authentifier la partie qui contient ce bloc. Les conditions d'accès déterminent laquelle des 2 clés doit être utilisé.



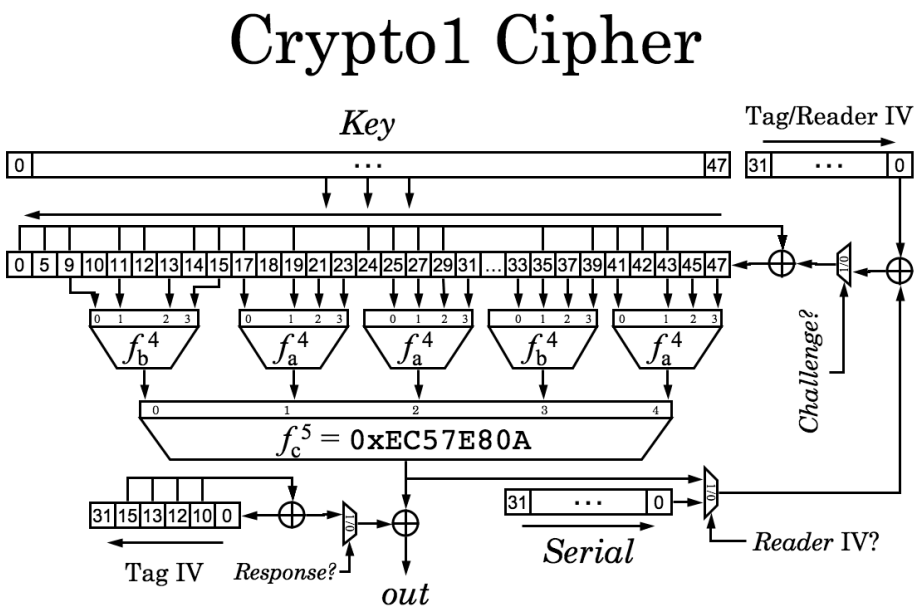
Mémoire d'une puce Mifare Classic

CRYPTO1

Après l'authentification, la communication entre le lecteur et la carte est chiffré avec la méthode de chiffrement CRYPTO1.

Ce chiffrement consiste en un registre à décalage à rétroaction linéaire (LFSR: Linear Feed-back Shift Register) de 48 bits avec la génération de polynôme ***x48 + x43 + x39 + x38 + x36 + x34 + x33 + x31 + x29 + x24 + x23 + x21 + x19 + x13 + x9 + x7 + x6 + x5 +1*** et une fonction *f* de filtrage non linéaire.

A chaque tour d'horloge, 20 bits du LFSR passent dans la fonction de filtrage, générant 1 bit de la clé. Ensuite le LFSR fait un décalage d'un bit vers la gauche et génère un nouveau bit sur la droite à l'aide du polynôme généré.



Le nonce

La puce Mifare Classic possède un générateur pseudo-aléatoire. Le tag nonce de 32 bits est généré par un LFSR de 16 bits avec le polynôme suivant : ***x16 + x14 + x13 + x11 + 1***.

A chaque tour d'horloge, le LFSR décale les bits vers la gauche et le nouveau bit est calculé.

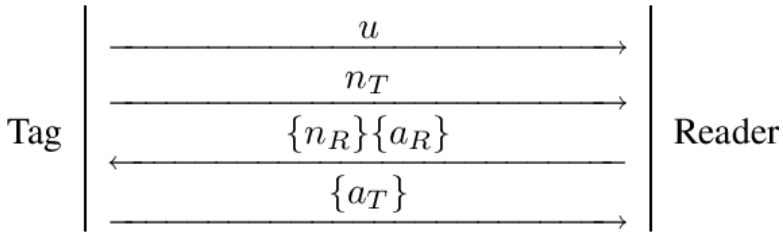
Sous les mêmes conditions (la puce et le lecteur n'ont pas été bougé), le nonce que la puce va générer ne dépends que du temps entre le déclenchement du champ électromagnétique du lecteur et le moment où celui-ci envoi le demande d'authentification.

En pratique, cela signifie qu'un attaquant ayant un contrôle physique sur la puce peut faire envoyer le **même nonce à chaque fois par la puce**.

Protocole d'authentification

Pendant la phase anticollision, la puce envoie son UID au lecteur. Il va ensuite va ensuite faire une demande d'authentification pour une partie de la puce particulière. La puce va alors envoyer un challenge nT. A partir de la communication sera cryptée.

Le lecteur réponds ensuite avec son propre challenge *{nR}* et la réponse *{aR}:=suc^{64}({nT})* au challenge de la puce. Enfin la puce finit avec sa réponse *{aT}:=suc96({nT})* au challenge du lecteur.



Protocole d'authentification

Les vulnérabilités

La Mifare Classic envoie un bit de parité pour chaque octet transmis. En ne respectant pas les standards, la Mifare Classic mélange la couche des données avec celle qui sécurise la communication : Les bits de parité sont calculés sur le texte a envoyé plutôt que sur les bits qui sont réellement envoyé (le texte chiffré). Ce qu'il se passe donc est en réalité : authentifier puis chiffrer, et cela n'est en général pas sécurisé.

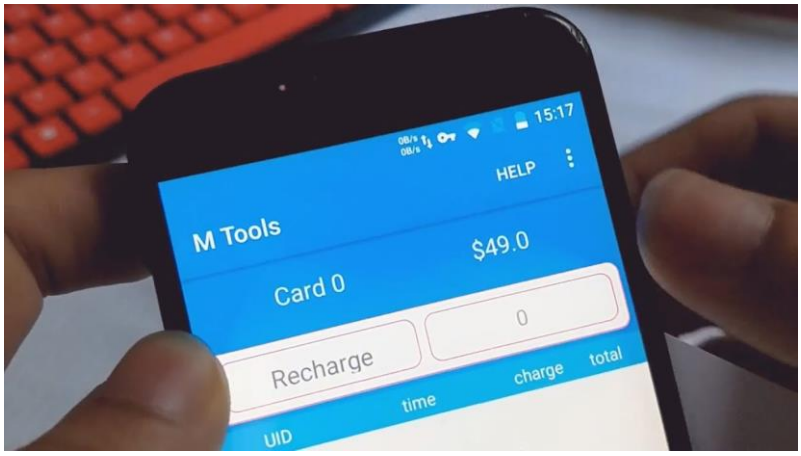
De plus, les bits de parités sont chiffrés avec le même bit de la clé qui chiffre le premier bit du prochain octet du texte. Pendant le protocole d'authentification, si le lecteur envoie les mauvais bits de parités, la puce va arrêter de communiquer. Et si le lecteur envoie les bons bits de parités mais les mauvaises données d'authentification, la puce va répondre avec un code d'erreur (chiffré). Cela va détruire la confidentialité du chiffrement permettant ainsi à un attaquant d'établir un canal secondaire.

La mémoire de la Mifare Classic est divisée en plusieurs morceaux (16 pour la version 1k). Chacune d'entre elle ayant sa propre clé secrète de 48 bits. Pour effectuer une opération sur une partie spécifique, le lecteur doit d'abord s'authentifier en utilisant la clé correspondante. Quand un attaquant c'est déjà authentifier sur une des partie (en connaissant la clé de celle-ci) et essaye directement de s'authentifier sur l'autre (sans connaitre la clé), cet essai permet de faire fuiter 32 bits d'informations à propos de la clé secrète de cette partie.

Les attaques

En bref

Il existe **4 attaques** (des variantes de la brute force) qui exploitent les vulnérabilités présentées précédemment. Ces attaques ont pour but de retrouver les clés cryptographiques d'une carte Mifare Classic en ayant un accès uniquement contactless à ces cartes.



Ces attaques font des compromis entre les optimisations des points suivant :

- Le temps de communication (le temps que l'attaquant a besoin de communiquer avec la carte)
- Le temps de calcul hors ligne (le temps nécessaire pour calculer les clés cryptographiques en utilisant les données récupérer par la lecture de la carte)
- Le temps de pré-calcul (le temps de génération unique de tables statiques)
- La taille de disque nécessaire (pour le stockage des tables statiques)
- Des vérifications spéciales comme savoir si l'attaquant connait déjà une clé de secteur ou non.

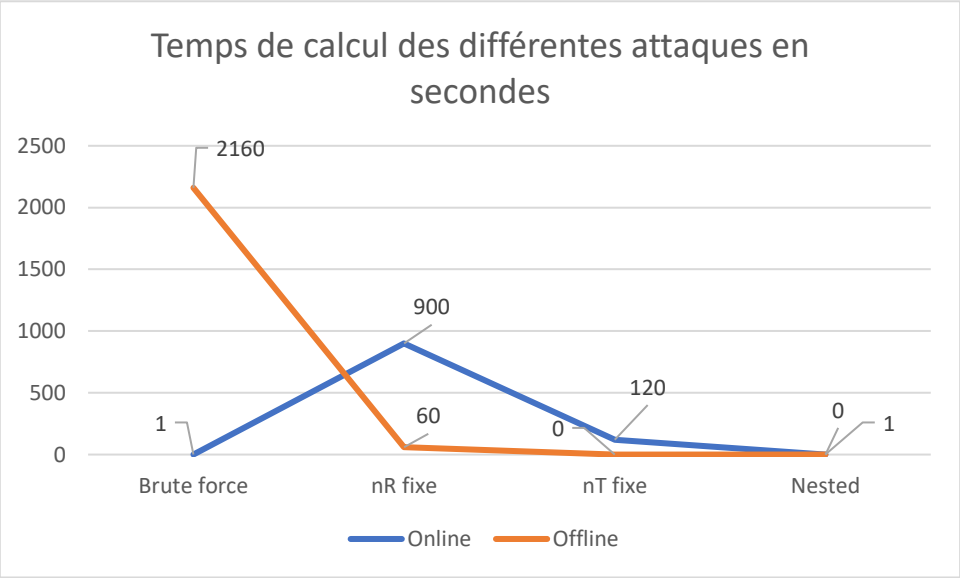
Voici la liste des différentes attaques :

- La première exploite la faiblesse des bits de parité afin de réaliser une attaque par force brute sur l'espace de clé de 48-bit. L'attaquant n'a besoin de s'authentifier approximativement seulement 1500 fois ce qui peut prendre moins d'une seconde (bien sûr il reste une partie de calculs hors ligne qui prends plus de temps).

- La deuxième attaque utilise la même vulnérabilité (le bit de parité). La différence est que cette fois c'est une attaque avec choix du ciphertext interactive (CCA2). Cette fois l'attaquant a besoin d'environ 28500 tentatives d'authentifications. Cette attaque consiste à s'assurer que le nonce aléatoire est constant sur la carte (ce qui fait monter le temps d'exécution à 15 minutes). Pendant ces tentatives d'authentification, l'attaquant choisit ses défis pour la carte de manière à obtenir un défi qui garantit uniquement 436 possibilités pour les bits impair du cipher interne. Ceci réduit la recherche à approximativement 33-bit et prends environ une minute.

- La troisième attaque consiste à garder le challenge constant (de l'attaquant) mais varier le challenge de la carte permettant d'obtenir encore une fois un état spécifique du cipher interne. Ces états spéciaux doivent être précalculés et sauvegardés dans une table de 384 GB. Cette attaque nécessite en moyenne 2^12 (soit 4096) tentatives d'authentifications ce qui prends en principe 2 minutes. Quelques tentatives d'authentification supplémentaires permettent une recherche plus efficace dans la table.

- La dernière attaque suppose que l'attaquant a déjà découvert une clé de secteur. En effet une fois que l'attaquant s'authentifie sur un secteur puis sur un autre, le protocole d'authentification est un peu différent. C'est-à-dire que le nonce de challenge n'est pas envoyé en clair mais crypté avec la clé du nouveau secteur. Parce que le générateur pseudo-aléatoire n'est que de 16 bits, mais aussi parce que le bit de parité fait fuiter 3 bits d'information et enfin que le générateur pseudo-aléatoire s'exécute de manière synchronisée avec le temps de communication, cela permet à un attaquant de deviner le texte clair du nonce de la carte et 32 bits du flux de clé. À cause des vulnérabilités du cipher il est possible d'utiliser ces 32 bits pour calculer environ 2^16 clés candidates. Ces clés peuvent ensuite être vérifiées hors ligne en utilisant une autre tentative d'authentification. Sachant que l'attaque nécessite uniquement 3 tentatives d'authentification le temps de communication est négligeable. La recherche hors ligne prend moins d'une seconde sur une machine ordinaire.



2 attaques en détail

Cette partie vise à montrer comment les vulnérabilités présentées précédemment peuvent être exploités.

Attaque par force brute

L'attaquant joue le rôle du lecteur et essaye de s'authentifier sur un secteur de son choix. Eve (l'attaquante) répond au défi de la carte avec 8 octets aléatoire ainsi que 8 bits de parité aléatoires pour *{nR}* et *{aR}*. Ceci donne une probabilité de 1/256 que les bits de parité soir correct et que le tag (la carte) réponde avec un code d'erreur crypté de 4-bit. Ce code d'erreur (un succès pour les bits de parité) fait fuiter 12 bits (sur 48) de la clé.

En répétant cette étape plusieurs fois (en général 6 fois est suffisant) cela permet de déterminer la clé. Sachant que la clé fait 48 bits de longueur, l'attaquante est en mesure de réaliser une attaque par force brute. Elle peut vérifier laquelle de ces 2^48 clés retourne les bits de parités corrects pour les 6 tentatives et garde la réponse. En pratique récupérer ces 6 tentatives d'authentifications avec des bits de parité corrects prends seulement en moyenne 6 * 256 = 1536 essais d'authentifications ce qui peut prendre **moins d'une seconde**.

Le temps nécessaire par la suite pour réaliser l'attaque par force brute hors ligne dépend de la machine à disposition de l'attaquant. Une estimation a été faite dans l'article utilisé[1] et prends exemple sur un calculateur d'une valeur de 10000$ en 2006. Ce dernier réalise des attaques brute force pour trouver des clés DES et en trouve une en moyenne au bout de 6.4 jours. Une estimation pessimiste conduit à dire que l'espace de clé de CRYPTO1 est 256 fois plus petit que celui de DES. On peut donc estimer le temps d'attaque par force brute sur la carte Mifare à 6.4 jours / 256 = 9216 / 256 = **36 minutes**.

Attaque en faisant varier le nonce du lecteur

Cette partie montre comment l'attaquant peut mettre en place une CCA (chosen ciphetext attack) en variant de manière interactive le cryptage de *{nR}*. On suppose ici que Eve peut contrôler le temps de mise en route du tag ceci afin que ce dernier donne toujours le même nonce *{nT}*.

Pour faire simple l'attaquante tente de déterminer les bons bits de parité. L'état interne du chiffrement de flux après avoir donné *{nR}* est noté α_{64}. Eve va par la suite refaire une session d'authentification en gardant les 31 premiers bits de *{nR}* ainsi que les 3 premier bits de parité et ensuite va inverser le dernier bit de *{nR}* et choisir le reste du nonce aléatoirement jusqu'à que la parité soit valide. À ce moment-là l'état du chiffrement de flux est $\alpha_{64} \oplus 1$ (α_{64} avec son dernier bit inversé). Puisque la parité du dernier octet de *{nR}* a changée (Eve a modifié uniquement le dernier bit) et puisque sa parité lors de la première exécution est cryptée par $f(\alpha_{64})$ et lors de la deuxième exécution par $f(\alpha_{64} \oplus 1)$ elle peut savoir si oui ou non le dernier bit de *{nR}* influence le cryptage du bit suivant si oui ou non $f(\alpha_{64}) = f(\alpha_{64} \oplus 1)$. Environ 9.4% des α_{64} possibles ont non $f(\alpha_{64}) \neq f(\alpha_{64} \oplus 1)$ et peuvent être générés facilement car seulement les 20 bits en entré de *f* sont pertinent. En répétant ses tentatives Eve peut après environ 10.6 essais trouver une instance de α_{64} faisant partie de ces 9.4% et finalement chercher hors ligne les 9.4% restant.

Conclusion

Comme nous l'avons démontré précédemment il est très facile de s'authentifier auprès d'une carte Mifare Classic dans des temps plus que raisonnables (maximum 36 minutes). Aussi les algorithmes comme la force brute sont à la portée de n'importe qui. Sachant que ces cartes sont encore utilisées dans de nombreux lieux comme le métro de Londres cette faille de sécurité doit être prise au sérieux. Aujourd'hui il existe un grand nombre de lecteurs permettant de dupliquer des cartes (faire des badges d'accès supplémentaires à la piscine de son immeuble par exemple). Encore plus facilement, certains téléphones proposant des lecteurs NFC peuvent aussi créer des duplicatas de carte d'accès très facilement ou en modifier les informations. Des cartes avec UID modifiable sont en vente et permettent, après réalisation des attaques présentés de copier le contenu des cartes ainsi que leur identifiant pour en faire des clones parfaits.

En termes d'attaques connues il a été découvert que les cartes du réseau souterrain de Londres fonctionnaient avec cette technologie et pouvaient être hackées facilement (tout comme celle des transports publics néerlandais). Ces deux institutions sont les plus connues mais sachant que le prix des cartes est extrêmement faible et facile à mettre en place on peut se douter qu'elles ont été utilisées en masse (10 milliards ont été vendu mais un grand nombre ont été détruites). Il est fort probable par exemple que les badges d'anciens immeuble soient en Mifare Classic ou encore certains bâtiments nécessitant un grand nombre d'accès (lycée par exemple).

De nos jours il existe la carte Mifare Plus qui utilise un chiffrement par block AES de 128 bits et qui est considéré comme sécurisé. Cependant le problème vient des lecteurs qui ne sont pas prévus pour ce type de chiffrement et beaucoup plus onéreux. C'est pour cela que la transition est relativement lente et que les Mifare Classic peuvent être encore utilisées à certains endroits.

Références :

1 : https://www.cs.ru.nl/~flaviog/publications/Pickpocketing.Mifare.pdf