NFC badges and security

Thierry Didi,
April 2025
Tidiwi

NFC (Near Field Communication) technology is now used in a large number of areas such as transport, electronic payment, access control in hotels, office or residential buildings, etc. The most commonly used badges are those of the MiFare^{TM}[(1)] family, designed by NXP Semiconductors. This document focuses on these badges that comply with ISO/IEC 14443 Type A specifications.

The idea to delve deeper into this subject came to me when I read a post on LinkedIn. The author describes how he was able to observe during a stay in a hotel that the Mifare Ultralight tags used to manage access to rooms were deployed outside of normal security recommendations, without the hotel manager being aware of it. As a result, this hotel could be the victim of theft from the rooms, by former residents or employees. As I read other posts, it became clear to me that these types of bad deployments, often to simplify the configuration and management of badges, were much more common than you might expect.

There are several families of Mifare TM badges (Mifare^{TM} UltralightTM, Mifare^{TM} ClassicTM, Mifare^{TM} PlusTM, Mifare^{TM} DesfireTM), and several generations in each of these families (EV1, EV2 ..) and I wondered which tags were used in which applications, and also how they were used. The underlying question was whether these tags provided a level of security appropriate for the resources they were intended to protect.

This document describes the current state of my research on this topic. In particular, he describes:

- The risks associated with the use of NFC badges.
- Attacks on NFC badges.
- The different attack scenarios.
- The tools available to attackers.
- The Mifare badge families, with a focus on security.
- Recommendations.

This document is not interested in the security of badge readers, which will be the subject of another study more broadly dedicated to the security of the IOT ("Internet Of Things").

It is aimed at integrators or users of solutions integrating NFC badges so that they can easily find their way around the different families of Mifare badges, while being informed of the risks associated with their choices.

As a preliminary remark, I would like to specify that the descriptions, recommendations, remarks contained in this document are my own and do not in any way commit NXP Semiconductors and that I have no affiliation with this company. This document is therefore based, beyond my personal experience, on:

- A search for the different tools (hardware and/or software) used in penetration tests ("Penetration tests" or "Pentests") and available in a completely legal way.

[1] The Mifare^{TM,} Mifare^{TM} ClassicTM, Mifare^{TM} PlusTM, Mifare^{TM} UltralightTM, Mifare^{TM} DesfireTM Trademarks are registered trademarks of NXP Semiconductors.

- Research on the various known attacks concerning NFC technology.
- Analysis of public datasheets of Mifare badges.
- Experiments with "Penetration Testing" tools on badges that I have in my possession.

If you have any comments, remarks or questions, do not hesitate to contact me by email to let me know: thierry.didi [at] tidiwi [dot] com

Introduction

Before going into the details of the technologies used in NFC, it is useful to specify a few elements:

- In this document, we are only interested in passive NFC badges, i.e. badges that are not equipped with a battery.

- An NFC badge first contains a unique identifier (7-byte encoded UID) or "non-unique" (4-byte encoded NUID[2]) identifier that identifies the badge, and by extension the badge bearer. This identifier is written by the badge manufacturer in an area of the memory that cannot be changed. We could therefore legitimately consider that we have identified a user once we have read the UID/NUID of his badge, and execute the actions requested by the user: for example, opening a building door. But this is actually a very bad idea because there are badges that are very easy to obtain, in a completely legal way, that allow you to clone this UID/NUID.

- An NFC badge is essentially composed of a non-volatile storage memory (EEPROM) in which information can be stored that is kept when the badge is not powered. The size of this memory is quite small and varies depending on the badge families, between a hundred bytes and a few thousand bytes for the most powerful badges.

- Beyond this memory, a badge contains an antenna and electronic components that allow it to generate a supply voltage when placed in an electromagnetic field generated by a "reader" and to perform cryptography operations. Once a badge has been "woken up" by a reader, the reader and the badge exchange a few messages to authenticate each other, then the reader can trigger actions on its environment (open a door, etc.) and/or modify the data stored in the badge (decrement the balance of an electronic wallet, a transport ticket, etc.).

- NFC technology is a "Radio Frequency" technology that operates at 13.56MHz. The distance at which a badge can be read by a reader is "normally" around ten centimetres. But a malicious user could use a reader whose output power would be greater than the authorized power, and equipped with a very large antenna (in a backpack for example): in this case, this reader could communicate with a badge that would be more than a meter away[3].

- This document focuses on NXP badges, although most of the issues identified can be easily transferred to other NFC badges. These badges are used in a large number of access control applications in companies, in residential buildings, in transport, in smart city projects... but they are not used (to my knowledge) in bank cards, in Parisian Navigo Passes or in electronic passports.

- The following acronyms and definitions are used in this document:

[2] The 4-byte encoded NUIDs can be reused by the badge manufacturer to identify multiple badges. However, the probability of having two badges with the same NUID in your possession is very low, even if it is not zero.

[3] https://www.rfidfuture.com/fr/rfid-read-range.html

- **AES**: is the symmetric encryption algorithm that is currently the most widely used in all systems aimed at ensuring the confidentiality or integrity of data. AES-128 is used to indicate that this algorithm is used with 128-bit encryption keys.
- **3DES**: is a symmetric encryption algorithm that is also widely used, but less robust than AES. It is gradually being replaced by the AES.
- **CMAC**: is an AES-based algorithm that generates a message signature from the message content and an AES key. During a communication, this signature is calculated by the sender, added to the end of the message and transmitted to the receiver. The receiver recalculates the signature of the received message, and verifies that it obtains the same signature as the one calculated by the sender. If it doesn't, it indicates that the message was modified by a malicious actor (or by a transmission error) between it was sent and received. In this case, it ignores the message.
- **ECDSA**: Elliptic Curve Digital Signature Algorithm. This algorithm makes it possible to generate a signature of a message or file using a private key known only to the signer. A recipient will be able to verify this signature from a public key known to everyone.
- **UID/NUID**: A unique identifier (UID) or non-unique identifier (NUID) that is stored in the badge by the badge manufacturer. This identifier can be encoded on 4 bytes (NUID) or 7 bytes (UID) depending on the badges.
- **Attacker**: A malicious user who seeks to hijack the nominal usage of the system.
- **NFC reader**: a terminal that communicates with an NFC badge (for example, a payment terminal in a restaurant, or a badge reader at the entrance of a building). The term "reader" is misused because this terminal will often also be able to write information in the badge's memory.
- **Side Channel Attacks**: These attacks make it possible to discover encryption keys based on the time it takes for a badge to respond to a request ("*Timing Attacks*"), or by analyzing its current consumption during an encryption operation ("*Power Analysis Attacks*"). At first glance, these attacks seem to require the implementation of complex devices, but there is a *Timing Attack* that allows you to find the keys to a MiFare Classic badge from a simple PC by analyzing the time it takes for a tag to respond to certain messages [4].

[4] https://www.sidechannel.blog/en/mifare-classic-2/

The risks associated with the use of NFC badges

Before deploying an NFC-based system, it is important to be aware of the risks to which one is exposed, and to assess whether these risks pose a problem for the system. These risks include:

- **Identity theft**
  An attacker impersonates a legitimate user to gain access to a building or service, or to travel by charging another user...

- **Editing the content of a badge**
  For example, an attacker changes the number of trips he can make in a transport system.

- **Access to private content stored on the map**
  Like the name or address of a user, for example.

- **Denial Of Service**
  A legitimate user no longer has access to a resource that they should have access to. This can have a negative effect on the end user's opinion of the reliability of the deployed system, to the point of causing them to turn away from it.

- **People Tracking**
  By recording the dates and times of presentation of a badge's UID on different readers, the system operator can track the movements of the badge bearer.

Attacks on NFC badges

The most common attacks on NFC badges include:

- **Eavesdropping**: An attacker can spy on the communication between a badge and a reader. This attack is very simple since you just have to install a radio receiver near the tag and the reader. You can also use a device consisting of two NFC antennas connected by a cable ("range extender") to duplicate the communication between the badge and a reader, and spy on the conversation remotely. The attacker could then:

  - Read the badge UID and thus impersonate the badge owner in some access control systems that rely only on the badge UID. Although this practice is to be avoided, it is implemented in some low-cost access control systems.
  - Have access to confidential data contained in the badge if communications are not encrypted on the radio interface between the badge and the reader. For example, the Mifare Classic and Mifare Ultralight badges do not encrypt communications on the radio interface, whereas the Mifare Plus or Mifare Desfire badges can.
  - In some cases, guessing the encryption keys used to protect the data contained in the badge.

- **Jamming**: An attacker can jam all or part of the communication between a badge and a reader. This attack is also very simple to carry out, even if it is illegal. The consequences can be significant, especially if the user ends up losing confidence in the installed system.

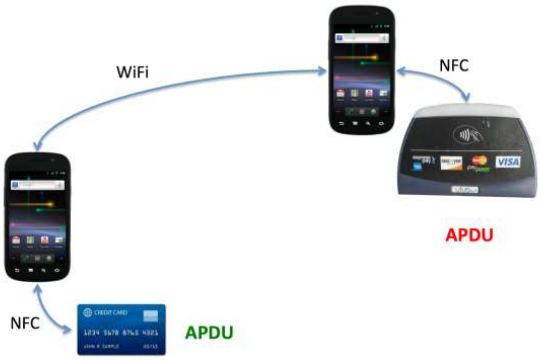
- **Reading and/or editing badge content**
  If the content of the badge is poorly protected, an attacker could read confidential data in the badge, or modify its contents, for example the number of trips they are allowed to take.

- **Replay Attack**
  An attacker eavesdrops on a conversation between a badge and a reader, and "replays" that conversation from an emulator. It could spy on the communication between an employee's badge and a badge reader at the entrance of a commercial building, then replay the same sequence in front of the reader to gain access to the building.

- **Badge cloning**
  Cloning a badge allows you to impersonate a legitimate user.

- **Using an Emulator**
  If the attacker has an emulator (which is very easy to obtain commercially, in a completely legal way), he can pass off this emulator as an authentic badge, as long as he has managed to read the content of the authentic badge.

- **Relay Attacks**
  This attack aims to make a badge communicate with a reader that is very far from the badge (for example in another country). Communications between the badge and the reader are relayed via a high-speed communication channel.



Source: https://salmg.net/2018/12/01/intro-to-nfc-payment-relay-attacks/relay

The data may be modified during transport. This type of attack would allow an attacker to gain access to a building by relaying the data of a legitimate user who is in a café or on the train, for example, or to execute contactless payments using another user's badge. There is an open source application (nfcgate[5]) developed to study the security of mobile applications that allows NFC data to be analyzed, modified or relayed between a badge and a remote server. This application has served as a basis for malicious users to develop applications capable of executing relay attacks.

- **Withholding of the Transaction Completion Message**
  If the attacker uses a relay between the device and its badge, they may decide to capture the completion message from the device, and not forward it to the badge. In this case, the badge

[5] https://github.com/nfcgate/nfcgate

data will not be updated because the badge will "think" that the transaction has been interrupted by the terminal before it is definitively validated. The terminal might not detect the fraud since, from its point of view, the transaction was completed successfully. For example, it could open a gate in a transport system, even though the user's badge has not been updated.

The different attack scenarios

Even if we imagine that an attacker has very sophisticated tools and great computer skills to compromise a system, this is not always the case. The scenarios to consider are as follows.

- An attacker attacks his own badge. It therefore has no time limit, and can use sophisticated tools to attack the badge. For example, they can try to change the number of tickets on their badge, or extend their access rights to a hotel room, or get access to rooms that are not their own...

- An attacker attacks a legitimate user's badge while they are sitting next to them on the train or in a coffee shop, for example. Time is not unlimited in this case. This scenario corresponds to a "Relay Attack", for example, or the theft of personal data. They can also read the badge's UID (or other data contained in the badge) that will allow them to impersonate the rightful owner.

- An attacker is trying to compromise another user's badge. The process is therefore more complicated, and involves recording conversations between the badge and a reader. Several conversations will often have to be spied on for the attack to succeed, which makes the process more complicated.

Tools available to attackers

To protect yourself from badge attacks, it's helpful to know what tools attackers might be using. This section focuses on very simple tools that can be obtained very easily and in a completely legal way. Beyond these tools, criminal or state organizations could have much more elaborate tools at their disposal.

- "Magic" badges where it is possible to configure a large number of parameters that are normally registered by the manufacturer (UID, Type of badge Mifare Classic 1K, or Mifare Classic 4K or Mifare Ultralight...).

- Software that allows you to manipulate Mifare Classic badge data, clone badges, or find the encryption keys used to protect the data of these badges (e.g. the MCT Android application).

- Electronic devices that allow you to emulate, read, write and attack Mifare Classic, Ultralight, Desfire badges (e.g. Flipper Zero, Chamelon Ultra, Proxmark 3).

- NFC antenna extenders ("Range Extenders") that allow, for example, to spy on exchanges between a reader and a badge. They are composed of two NFC antennas connected by an RF cable. These antenna extenders can also be discreetly integrated into a compromised NFC reader, so that the badge presented on the reader communicates with a remote and hidden reader without the owner's knowledge.

- Software derived from software used for cybersecurity research that makes it possible to relay radio signals between a local reader and a remote reader, which could even be in another country.

Mifare badge families

Over time, NXP has introduced several badge families that are compatible with (or based on) the ISO/IEC 14443 Type A family standards [6]. Each family meets the needs of a certain level of security. In each tag family (Mifare Classic, Mifare Ultralight, Mifare Plus, Mifare Desfire), the specifications have evolved and it is recommended to use the latest tags in new designs. However, many of the systems deployed are based on the older, and therefore least secure, versions of these tags. It is therefore useful to know the characteristics of these older versions if you are interested in the security of the systems deployed.

These families are as follows:

- **Mifare Classic**: this is undoubtedly the most used badge, and has been the subject of a large number of studies and security attacks. It uses a proprietary encryption algorithm named Crypto1. But this algorithm has been completely broken, and there is now a lot of software that can hack these badges. Despite these shortcomings, these badges are still widely used to secure access to hotel rooms, at the entrance to residential or tertiary buildings, or to authorize access to public facilities (swimming pools, waste collection centers, etc.), or to manage access to events. Originally, they were also widely used in transport.
  The specification of Mifare Classic tags has evolved over time:
  - **Mifare Classic 1K / Mifare Classic 4K** : the oldest
  - **Mifare Classic EV1** : NXP's current recommendation for Mifare Classic badges.

- **Mifare Plus**: Mifare Plus badges were developed as an alternative to Mifare Classic badges, offering security based on the AES algorithm rather than Crypto1. So they offer much better security, but I'm not sure they're widely deployed today. The changes to the Mifare Plus badges are as follows:
  - **Mifare Plus SE**
  - **Mifare Plus EV2**

- **Mifare Ultralight**: These badges are often used to secure access to hotel rooms, for example, or to manage single-use or limited-use tickets, or to control access to events (concerts, etc.). There are currently 4 versions:

  - **Mifare Ultralight:** This version is now deprecated but widely deployed and without any security features.
  - **Mifare Ultralight EV1:** This version offers "ultra-minimalist" security based on a 32-bit password.
  - **Mifare Ultralight C:** This version offers better security based on 3DES encryption.
  - **Mifare Ultralight AES:** This version is intended to replace Mifare Ultralight C. It uses the AES encryption algorithm and incorporates new security features.

- **Mifare Desfire**: These badges offer the highest level of security, and larger memory sizes than other badges. They are intended for applications that require a high level of security, and due to their large memory, they can be used to manage access to several different services with a

[6] All MIFARE tags are compatible with ISO/IEC 14443 Part 2 and ISO/IEC 14443 Part 3 – MIFARE Desfire and MIFARE Plus tags are compatible with ISO/IEC 1443 Part 4 – MIFARE Classic and MIFARE Ultra light tags are compatible with the MIFARE protocol, which is based on ISO/IEC 14443 Part 3.

single badge. In addition, the access control systems for tertiary buildings that used to use Mifare Classic are gradually migrating to Mifare Desfire which offers a much higher level of security.

  - **Mifare Desfire EV3**

Each of the families mentioned above has evolved over time, most often to respond to the security vulnerabilities identified. Often, these evolutions are translated into the trade name of the badge by adding the evolution number, in the form EVx. For example, the Mifare Desfire tags existed as Desfire EV1, then Desfire EV2, and now Desfire EV3.

Mifare Classic badges (MF1S50yyX)

Mifare Classic badges are the oldest NFC badges developed by NXP, originally for transport solutions. They are now obsolete.

These badges are however widely used in a large number of access control applications (for example, these badges are used in France in the VIGIK system which manages access to a large number of residential buildings [7]), transport, but also in a large number of applications with no link to security. The level of security offered by these badges is almost zero.

- **THE UID/NUID**
  Each badge has a unique and non-modifiable identifier (UID) consisting of 7 or non-unique and non-modifiable (NUID) consisting of 4 bytes. This identifier is stored in sector 0 of the badge's memory which can be read by an NFC reader without requiring prior authentication.

- **Memory**
  These badges can have a memory of 1k bytes (Mifare Classic 1K) or 4k bytes (Mifare Classic 4K). The memory of the Mifare Classic 1K badges is organized into 16 sectors of 64 bytes each. Insofar as each sector has its own access control keys (see below), it is possible to dedicate each sector to a specific application (e.g. access control, ticketing, etc.).

- **Data protection**
  Each sector is associated with a key pair (KeyA/KeyB) that are used to protect read and/or write access. The data in each sector can be protected by these keys and thanks to a symmetric encryption algorithm called Crypto1. Since this algorithm is a symmetric algorithm, the encryption keys for a sector of a badge must be known by the badge AND by the reader in order for the reader to access that sector.
  It should be noted that the badges come with default keys, which are publicly known. It is therefore essential to customize these keys if confidential information needs to be stored in these sectors (for example, the number of passages allowed through a metro gate).

- **Remarks**

  - Different sets of keys should be used for each badge, to prevent a compromised badge from compromising the entire system. A best practice recommended by NXP is to derive a set of keys for each badge from a Master Key and fixed information such as the badge's UID. This Master Key will only be known to the reader and will not be stored in the badge.

  - One might be tempted to use the UID of a badge to identify the badge user, which is done by many apps. However, this method is not at all secure because it is possible to buy badges ("Magic Cards") whose sector 0 is modifiable: it is therefore quite

[7] https://www.vigik.com/presentation-de-vigik

possible and very simple to make a new badge that would have the same UID as the original badge. This is even easier if you use a badge emulator.

    - It should be noted that there is also open source software that can execute a dictionary attack on the Crypto1 algorithm to retrieve the key of a sector of a Mifare Classic badge. These software programs first test the default keys, or keys that have been published by individuals and are freely available on the web (for example in the source files of the MifareClassicTool application).

    - There are also other types of attacks that are also based on flaws in the Crypto1 algorithm, in particular the MFKEY32 attack ("mfkey32 Attack") to find the Key A and Key B keys of a sector from the capture and analysis of a few authentication exchanges between the reader and the badge.

    - Finally, we might be tempted to customize only the keys of the sectors we use, and to leave the keys of the other sectors at their default values. However, there is a "Nested Attack" linked to flaws in the Crypto 1 algorithm that allows you to retrieve the keys of all sectors of a Mifare Classic badge if you know at least the key of a sector.

Mifare Classic EV1 (MF1S50YYX_V1)

These badges are the most recent version of the Mifare Classic badges. Their features are essentially the same as those of the Mifare Classic badges. The most notable difference is as follows:

- **Proof of origin**
  Mifare Classic EV1 tags support a verification of their origin function, which allows a reader to verify that it is talking to a genuine tag. This verification is based on an ECDSA signature generated by NXP from a private key known only to NXP, and stored in the badge. A reader who has the corresponding public key will be able to verify this signature.

- **Remarks**
    - These badges therefore suffer from the same weaknesses as the Mifare Classic badges.
    - The origin verification feature provides an excellent level of protection against the creation of fake badges with a new UID since an attacker will not be able to generate an authentic signature of this new badge.
    - This function is obviously only usable if the player supports it.
    - It also protects against the cloning of a badge on a blank badge.
    - However, it does not protect against cloning an existing badge using an emulator. This emulator will be able to spy on the communication between a reader and the original badge when its signature is verified, and then reproduce the answers of the authentic badge when it is in front of the reader.

Mifare Plus SE (MF1SEP(H)10x1)

Mifare Plus SE badges have been developed to correct security vulnerabilities in Mifare Classic badges and to enable migration from a Mifare Classic infrastructure to a more secure infrastructure. Notably, they use the AES-128 algorithm for encryption as a replacement for the Crypto1 algorithm. These badges have been designed to be compatible with an infrastructure deployed for Mifare Classic badges. They therefore potentially contain two sets of keys: a Crypto 1 keyset and an AES-128 keyset. Therefore, they must also implement the Crypto1 algorithm, and are therefore subject to the attacks mentioned above when used with legacy infrastructure.

- **THE UID/NUID**
  Each badge has a unique non-modifiable identifier (UID) consisting of 7 or a Non-Unique and non-modifiable identifier (NUID) consisting of 4 bytes.

- **Memory**
  The memory organization is the same as the memory organization of the Mifare Classic badges.

- **Data protection**
  - During mutual authentication between the badge and the reader, two AES keys (session keys) are generated. These keys are therefore different for each new session, which protects against Replay attacks. These keys will be used to encrypt the data on the radio interface, and to ensure the integrity of these messages.

  - Sectors can be protected by the Crypto1 algorithm and a Key A/Key B key pair as in the case of the Mifare Classic badges. They can also be protected by the AES-128 algorithm and an AES-128 key pair. AES keys can be updated.

- **Proof of origin**
  To protect against cloning, these badges offer a function that allows you to prove that they are indeed original badges. This function is based on a special AES key, dedicated to this feature. The reader must therefore know this AES key to verify the origin of the badge.

- **Remarks**
  These badges provide an excellent level of security against most attacks, unless of course they are used on legacy infrastructure with data protected by the Crypto 1 algorithm.
  They are protected against replay attacks.
  They remain subject to relay attacks, and to tracking people by means of the badge's UID/NUID.

Mifare Plus EV2 (MF1P(H)x2)

Mifare Plus EV2 badges offer a level of security compatible with the needs of banking applications, or with electronic passports.
The Mifare Plus EV2 badges are also compatible with the Mifare Plus SE badges and the Mifare Classic EV1 badges. In addition, they have additional features. In particular, a proximity detection function between the card and the reader, to protect against relay attacks. Finally, the messages exchanged

between the card and the reader can be encrypted or authenticated using two AES keys. These keys are generated when the badge is authenticated by the reader and are different for each session. This helps protect against replay attacks.

- **UID/NUID**
  - Each badge has a unique non-modifiable identifier (UID) consisting of 7 or a Non-Unique and non-modifiable identifier (NUID) consisting of 4 bytes.
  - This badge can also be configured to generate a "pseudo-random" UID each time you log in. This feature protects the anonymity of the wearer: the system manager will not be able to trace the user's movements by following the appearances of his badge on the various readers of his installation.

- **Memory**

  The memory of these badges is composed of 2048 or 4096 bytes.

- **Data protection**
  - The badge data can be protected against read and/or write access. This protection can be achieved by the Crypto1 algorithm and a KeyA/KeyB key pair if the badge is used against an infrastructure deployed for Mifare Classic badges (therefore with a low level of security), or by the AES-128 algorithm with AES keys in the case of a recent infrastructure.
  - Radio Interface Protection

    Radio communications between the reader and the badge can be encrypted with the AES-128 algorithm, and their integrity can be verified by adding a CMAC signature to the end of each message.
- **Proximity Check**

  This badge implements a proximity verification feature, based on the precise measurement of the transit time of messages between the reader and the badge. This feature must be initiated by the reader (so the reader must support it).

- **Proof of origin**

  These badges offer two methods to verify that they are original:
  - An ECDSA signature computed using a private key known only to NXP. The reader must know the public key associated with that private key to perform signature verification.
  - An AES128 key (which must therefore be known to the badge and the reader).

- **Remarks**

  These badges help protect against all known attacks.

  The "Proximity Check" feature probably protects well against relay attacks when the transit time is "important" (a few milliseconds) between the reader and the badge. This is the case if the reader and the badge are located in different places, and the signal is carried via the WIFI network, or 4G ... On the other hand, this feature probably does not offer protection if the reader and the badge are separated by a "*range extender*" because the signal transit time on this type of device is very short.

Mifare Ultralight (MF0ICU1)

🛑

This badge is no longer referenced on the NXP website, but it can still be purchased on the internet and is still used in applications.

- **UID**
  The UID of this badge is composed of 7 non-modifiable bytes.
- **Memory**
  - The memory of this badge contains 48 bytes (384 bits) that can be used to store data. Unlike the mifare Classic badges, this memory is not partitioned into disjoint sectors.
- **Protection**
  - No authentication is required to access the data, and this badge does not have any encryption features.
  - Each page can be locked to become "Read Only" irreversibly.
- **Remarks**
  - This badge should never be used in security applications since it is very easy to clone it with an emulator, although I was able to verify that it is currently used in home alarm systems.

Mifare UltraLight EV1 (MF0ULX1)

- **UID**
  The UID of this badge is composed of 7 non-modifiable bytes.
- **Memory**
  The memory of this badge contains 48 bytes (384 bits) or 128 bytes (1024 bits) that can be used to store user data.
- **Protection**
  - Each page can be locked to become "Read Only" irreversibly.
  - Memory access (read and/or write) can be protected by a 32-bit password
  - It is possible to program a maximum number of authentication failures to protect against brute force attacks.
- **Proof of origin**
  This badge provides a feature to ensure that it is an original badge and not a copy. The badge contains an Elliptic Curve Digitla Signature Algorithm (ECDSA) signature generated by NXP from a private key known only to NXP. This signature can be verified by a reader who has the public key associated with that private key. The advantage of using this type of signature is that the private key used to generate the signature is not known to the badge or the reader. This insurance protects against duplication of an original badge on a blank badge, but not against duplication by an emulator that could also reproduce the signature of the original badge.
- **Other features**
  This badge implements 3 "one way counters", i.e. these counters can only be incremented by a reader, and never decremented. These counters can be used to limit the number of trips in means of transport, for example.

- **Remarks**
  To access the badge data, the reader starts by sending the badge password "in plain text" to unlock it... So, an attacker listening to the communication could easily know the password to the badge. The security offered by this badge is therefore ultra-minimalist. Some implementations use the same password for all badges, which turns out to be heresy [8]. When using this type of badge, it is essential to use different passwords for all badges, and to configure a maximum number of authentication failures to protect against brute force attacks.

Mifare UltraLight C (MF0ICU2)

🛑

This badge is similar to the Mifare Ultralight EV1 badge, but it implements better memory protection. However, NXP now recommends using the Mifare UltraLight AES badge instead (see below) which implements a more robust encryption algorithm and other security-enhancing features.

- **UID**
  The UID of this badge is composed of 7 non-modifiable bytes.
- **Memory**
  The memory of this badge contains 144 bytes (1152 bits) that can be used to store user data.
- **Protection**
  - Memory can be locked (in blocks) to become "Read Only" irreversibly.
  - Access to the memory can be protected by mutual authentication: the badge and the reader must prove to each other that they know the same 3DES encryption key. This exchange is based on the 3DES algorithm in CBC (Cipher Block Chaining) mode. This mechanism is more effective than password protection because the encryption key that will be used to access the memory is a key derived from this shared key, which is itself never transmitted in clear text over the NFC interface.
- **Other features**
  - This badge implements a "one way" counter that can be incremented but never decremented.
- **Remarks**
  Even though the level of security is higher than for the Mifare Ultralight EV1 badges, this badge still does not offer absolute security. The 3DES algorithm is considered quite weak, and its implementation in badges, especially the CBC mode with a zero IV (Initialization Vector) makes it not very robust against cryptographic attacks. But this type of attack does not seem to be implemented yet in penetration testing tools.
  In all cases, it is strongly recommended to use unique encryption keys for each badge. Since the reader needs to know the keys to all badges, it is not possible to randomly generate keys for each badge. Instead, a "derivation function" will be used to generate a unique key from the badge UID and a "Master Key". This Master Key will be known by the factory that personalizes the badge and by the reader, but it will not be stored in the badge.

[8] https://www.linkedin.com/posts/stephanelemee_vuln%C3%A9rabilit%C3%A9-s%C3%A9curit%C3%A9-d%C3%A9poussi%C3%A9rage-activity-7127927457076109312-o7m8/?originalSubdomain=fr

Mifare UltraLight AES (MF0AES(H)20)

This badge is an evolution of the Mifare Ultralight C badge. The cryptography is based on the AES-128 algorithm which is more robust than 3DES. It also adds message authentication on the NFC interface to protect the system against message changes between the reader and the badge ("Man In The Middle" attacks).

- **UID**
  - The UID of this badge is composed of 7 non-modifiable bytes.
  - This badge can also be configured to generate a "pseudo-random" UID each time you log in. This feature protects the anonymity of the wearer: the system manager will not be able to trace the user's movements by following the appearances of his badge on the various readers of his installation.
- **Memory**
  The memory of this badge contains 144 bytes (1152 bits) that can be used to store user data.
- **Protection**
  - Memory can be locked (in blocks) to become "Read Only" irreversibly.
  - Access to the memory can be protected by mutual authentication: the badge and the reader must prove to each other that they know the same AES-128 encryption key. This exchange is based on the AES algorithm in CBC (Cipher Block Chaining) mode.
  - Messages can be authenticated on the radio interface by adding a signature calculated with the CMAC algorithm.
  - The messages are not encrypted on the radio interface: they are transmitted in the clear.
  - It is possible to limit the number of successive authentication failures. If the number of authentication failures is exceeded, the badge is permanently rendered unusable.
- **Proof of origin**
  This badge provides a feature to ensure that it is an original badge and not a copy. This insurance protects against duplication of an original badge on a blank badge, but not against duplication by an emulator that could also reproduce the signature of the original badge.
- **Other features**
  This badge implements 3 "one way counters", i.e. they can only be incremented by a reader, and never decremented. These counters can be used to limit the number of trips in means of transport, for example.
- **Remarks**
  - Even though the AES algorithm is more robust than the 3DES algorithm, this badge also uses the CBC mode with a fixed IV (Initialization Vector), which makes it vulnerable to cryptographic attacks. But this type of attack does not seem to be implemented yet in penetration testing tools.
  - As with the Mifare Ultralight C badge, it is strongly recommended to use unique encryption keys for each badge (see Mifare Ultralight C tag).

Mifare Desfire EV3 (MF3D(H)x3)

This badge is the most recent version of the Mifare Desfire family, after Mifare Desfire EV1 and Mifare Desfire EV2. It is the badge with the highest level of security of all Mifare badges, which is the same level as recommended for payment cards or ePassports. NXP recommends its use in smart city applications, where the same badge could be used to access different services (transport, vehicle sharing, micro-payment, etc.). Each application can store its own files on it, within the limit of the available memory size. The memory size varies from 2kBytes to 16kBytes depending on the model.

These badges provide a badge originality check feature, and implement proximity detection features to protect against relay attacks.

- **UID**
  - The UID of this badge is composed of 7 non-modifiable bytes.
  - This badge can also be configured to generate a "pseudo-random" UID each time you log in. This feature protects the anonymity of the wearer: the system manager will not be able to trace the user's movements by following the appearances of his badge on the various readers of his installation.
- **Memory**

  The memory of this badge contains 2kBytes, 4kBytes, 8kBytes, or 16kBytes that can be used to store user data. This memory is managed by a File System, where applications can create files. This organization is flexible so that the badge can house multiple applications with different storage needs.
- **Protection**
  - Memory is partitioned so that an application cannot access another application's data unless it is explicitly allowed to do so.
  - At the badge level, a number of 3DES and/or AES keys protect access to the badge. The owner of these keys (i.e. the badge system manager) does not have access to the application data.
  - Each application can set a number of encryption keys (3DES or AES) to protect the read and/or write files that belong to it.
  - Application data can be encrypted and authenticated on the radio interface using keys that are unique to each application.
  - Encryption keys can be updated.

- **Proof of origin**
  - This badge provides a proof-of-origin function based on an ECDSA signature generated by NXP.

- **Proximity Check**
  - This badge implements a proximity check feature to protect against relay attacks. This feature is based on a very accurate measurement of the transit time of messages on the radio interface between the reader and the badge. This feature must be initiated by the reader (so the reader must support it).
  - This feature probably has the same weakness as for the Mifare Plus badges if the reader and the badge are connected via a "*range extender*", i.e. a pair of NFC antennas connected by an RF cable.

Recommendations

This section lists the most important recommendations to follow when deploying an NFC-based system. As discussed in the introduction, these recommendations only apply to badges and not to readers. Since all badge families use symmetric encryption algorithms (Crypto1, 3DES, or AES) to protect data, readers know or know how to derive the keys to all badges. The compromise of a reader is therefore an absolutely critical element. The security of readers will be the subject of a separate study as part of the security of the IOT.

- First of all, as recommended in "Secure By Design" practices, a risk study must be carried out. The objective of this study is of course to identify the risks to which we are exposed, but above all to define whether or not these risks are acceptable for the system we are deploying. This risk analysis will make it possible to select the badge family compatible with these risks. Not all apps need Mifare Desfire EV3 badges...

- Limiting the effects of an attack
  Even though badge designers are introducing more and more security-related features, it is possible, even likely, that attackers will one day be able to bypass them. It is therefore necessary to implement mechanisms at the system level to detect abnormal behavior, for example the use of the same badge in two different places at very close times, or the abnormal evolution of the number of passages authorized in a transport badge...

- Never use the UID/UUID of a badge to perform an action as this UID/UUID is easily clonable.

- Key diversification
  Using a different set of keys for each badge ensures that if a badge is compromised, an attacker will only be able to modify the data of a badge with the same UID: that is, that same badge, or a clone of that badge. This diversification could be done by deriving a badge's key set from its UID and a Master Key that itself is not stored in the badge.

- Encrypt the data stored in the badges, using a robust algorithm (AES for example) and encryption keys known only to the reader, i.e. keys that are different from the keys used to access the badge sectors.

- Implement Black Lists and/or White Lists in terminals. The use of Black Lists makes it possible to complete the key diversification mechanism by banning a UID that has been compromised. The use of White Lists is more secure, but it is complex to implement if the system is composed of a large number of badges.

- Implement a MAC on badge content
  Each time a terminal writes data to a badge, it calculates a MAC code from this data and the badge's UID. This MAC is computed from a key that is known only to the reader, and which can itself be derived from a Master Key. But this does not prevent an attacker from restoring the adapter to a previous state whose MAC is known. So this mechanism is inefficient in a public transport application that would count the number of passages of a user for example.

- The Mifare Plus and Mifare Desfire badges allow you to update the keys of a badge. If the Master Key is compromised, a new Master Key can be deployed in all terminals, and some terminals that are in secure areas can be programmed to reprogram the badge key sets from this new Master Key.

- Securing the key personalization process
  Since security is based on symmetric encryption algorithms, these keys are programmed into the badges when the badges are personalized. It is therefore necessary to ensure that the premises and the process used for this customization are properly secured.

ABOUT TIDIWI

TIDIWI is a consulting company specializing in the design and development of IoT projects. In particular, TIDIWI has strong expertise in wireless communication protocols and ultra-low-power systems.
TIDIWI provides consulting services, embedded software development, electronic board design, development team management, and project management.

For more information, visit TIDIWI's website: www.tidiwi.com

Or contact: thierry.didi[at]tidiwi[dot]com

References

- AN10969 : System level security measures for MIFARE installations
- AN12653 : End to end system security risk considerations for implementing contactless cards and tags
- MF0ULX1 - MIFARE Ultralight EV1 - Contactless ticket IC
- MF0ICU2 - MIFARE Ultralight C - Contactless ticket IC
- MF0AES(H)20 - MIFARE Ultralight AES contactless limited-use IC
- MF1S50YYX_V1 - MIFARE Classic EV1 1K - Mainstream contactless smart card IC for fast and easy solution development
- MF1SEP(H)10x1 - MIFARE PLUS SE - Secure contactless smart card IC for seamless migration
- MF1P(H)x2 - MIFARE Plus EV2
- MF3D(H)x3 - MIFARE DESFire EV3 contactless multi-application IC