Developing Countermeasures against Cloning of Identity Tokens in Legacy Systems

Pavel Moravec, Michal Krumnikl

Developing Countermeasures against Cloning of Identity Tokens in Legacy Systems

Pavel Moravec[1,2] and Michal Krumnikl[1,2]

[1] Department of Computer Science, FEECS,
VŠB - Technical University of Ostrava,
17. listopadu 15/2172, 708 33 Ostrava-Poruba,
Czech Republic
[2] IT4Innovations, VŠB - Technical University of Ostrava,
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
{pavel.moravec, michal.krumnikl}@vsb.cz

Abstract. During the development of a new access system based on modern RFID technologies it was found that companies producing access control systems for residential and office buildings still prefer the use of existing cheap solutions instead of incorporating new technologies. This is mainly due to the additional costs new systems require.
The used legacy technologies are however prone to identity token cloning which allows easy access of unauthorized people to buildings. In previous paper, we have already briefly described a way how to detect cloned RFID tokens in 125 kHz RFID system.
This paper lists the risks of the legacy access systems and offers ways how to detect a cloned 125 kHz tag, 13.56 MHz RFID MIFARE Classic card or Dallas Semiconductors iButton access token and how to pro-actively disable them.

Keywords: RFID;legacy access systems;identity token;DS1990;EM4100;MIFARE Classic;cloning

1 Introduction

During the last year, we have developed an access system using the state-of-the-art technologies. This however meant that the production costs of the final device exceeded $ 50 mark. The company we have done this research for then tried to offer the solution to their partners as a drop-in replacement for legacy modules, but was not successful due to the production costs being considered too high. The companies still prefer the use of legacy solutions which are known to be unsafe, and users may create clones of the access tokens in a relatively simple way. This means that the physical security of such buildings, lifts or restricted areas is compromised and the access may be presently gained by making a cheap clone of an access token, sometimes requiring as little as taking a photograph of the token or reading the RFID token wirelessly and creating its clone later on.

Nowadays, there is a huge variety of tags and labels used in modern access systems. Tags can be either passive or active and they differ in size, casing, storage capacity and

used microcontrollers. The functions they provide range from transmitting read-only factory-assigned serial number to bi-directional communication with high-end cryptographic features.

Most electronic access systems use radio-frequency identification chips (RFID) as they provide simple use for end users. As the number of RFID applications increases, attacks against both tags and readers are getting more and more frequent. We may say that the price of solution usually reflects the level of security. Nevertheless, even more expensive RFID solutions do not always guarantee higher level of security. In the past, MIFARE Classic provided sufficient protection against attacks for reasonable price. Despite the fact that they are nowadays considered insecure and there are several easy to implement attacks against them [7,10], they are still being used. An improved and more secure version, MIFARE DESFire tags were broken few years later using a non-invasive power analysis and template attacks [18].

In other tokens, like HID iClass similar flaws were found [5]. Insecure RFID technologies were also found in Hitag2 car keys [20]. More practical approaches of eavesdropping, unauthorized scanning and relay attacks on ISO 14443-A tokens and radio layer were described in [8,9,14].

Wired technologies, like iButtons are vulnerable to similar attacks. One of the first hacking attempt on iButtons was presented on CCC Conference [2] followed by other presentations on this topic [6]. Secure iButtons that are using a SHA-1 enabled DS1963S chips were broken as well [3]. In order to maintain the same level of security over the years, manufacturers would have to replace the broken technologies in their installations. This is expensive and hard to manage.

Our goal is to provide a cheap drop-in module for present solutions, which will detect – and disable – cheap clones without their user gaining physical access to the building or restricted area. The resulting solution must be cheap (less than $ 10), provide a drop-in replacement for original module and be able to detect at least the commercially available tokens used for cloning. Ideally, the solution may also combine some of the legacy technologies, to provide an added value for the company whilst being compatible with the original module.

The paper will be organized as follows: First, we will describe the technologies with access tokens, which are still being sold for new installations. Second, we will discuss the possibilities of cloning such tokens and commercially available solutions for token cloning and their prices. Third, we will discus the possibilities of detecting these commercially available counterfeit access tokens and provide a framework to disable them. Further, we will offer solutions for individual technologies and provide the costs and comparison with the original solution.

2 Electronic Access Control Systems

In this section, we will discuss the electronic access control systems with identity token technologies used mainly in residential buildings, which are presently sold in Czech Republic. There are generally two areas:

– contact system sold nowadays solely with support of Dallas Semiconductors iButton Serial number tags of DS1990 [12] series,
– passive proximity cards offering contactless operation with the reader supplying power needed to operate the device. These systems are operating either on 125 kHz or 13.56 MHz band.

2.1 Dallas Semiconductors DS1990 iButton Serial Number

The iButton serial number (SN) is a widespread technology which is used for both gaining access to the residential buildings or restricted areas and for allowing special operation inside the lifts (access to some floors, operation of the lift only with the tag).

Each tag contains a unique 56-bit serial number and its 8-bit checksum, which the manufacturer presents as a 64-bit unique registration number. The chip is encased in a stainless steel case with the value of serial number laser-printed in a hexadecimal format on the positive pole of the case, which is typically mounted on a plastic keychain fob. The identification can be obtained through 1-Wire serial protocol by momentary contact. Communication with the bus master is performed by transmitting a 16.3 kbps signal on a wire used to power the device [12].

2.2 EM Microelectronic Passive Proximity EM41xx RFID Transponder

EM4100/4102/4105 tags are low frequency tags manufactured by EM Microelectronic. It is a RFID transponder containing 64 bits of read only memory programmed to store a 40-bit long unique ID. Remaining bits are used for the header and parity bits. The output is modulated using the Manchester coding on 125 kHz carrier with ASK modulation [4]. EM41xx tags are passive, drawing power from the RF field created by the reader. These tags cannot be rewritten, however there are other types that can be programmed to emulate original behavior.

2.3 Contactless RFID Tokens for Access Management

These tokens typically operate in 13.56 MHz frequency range and may be capable of storing user data. They either provide just a factory-set ID, or several kB of additional memory with read/write capability and ISO 14443 compliance. They use either a 4 byte non-unique ID or a 7 byte unique ID. Recently, tags with a 10 byte unique ID have been also introduced.

Each ISO 14443 Type A compatible tag with 7 byte ID should contain a 56-bit globally unique serial number [15], since each manufacturer is being assigned their own specific prefix. However, this does not take into account the situation, where counterfeit products exist on the market, duplicating the unique numbers assigned to legit manufacturers.

Typical example of such smart card ICs is the NXP Semiconductors MIFARE card family, which offer in their MIFARE Classic [16,17] variant either a 32-bit ID or unique 56-bit ID, and an EEPROM with 1kB to 4kB storage capacity. The ID is stored in the first 7 bytes of EEPROM memory, however for most functions we use only the

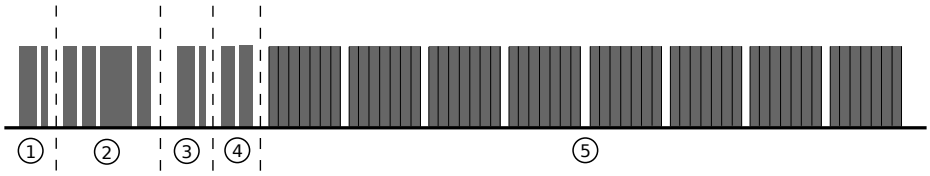

Fig. 1. RW1990 timing diagram during read and write operations [13]. Due to a different timing of write mode, the time axis is not to scale. The reader periodically checks the 1-Wire bus for the presence of iButton (1). When the iButton tag is detected, search ROM command is triggered. The tag responds with its family code, ROM code and CRC (2). Write operation is triggered after the reset (3), followed by a special command (4). Writing a new serial number is performed in a non-standard way. Each written bit is followed by 10 ms high state delay on the bus (5). The whole procedure is finished in about 680 ms.

first 4 bytes of the UID. These cards were superseded by NXP MIFARE DESFire card family in many applications, but remain still used in the legacy systems used for access management – especially where storing additional data on the card, e.g. the number of accesses, electronic waller, or time limit when the entry is permitted.

3 Identity Token Cloning using Commercially available Tags

Unfortunately, it is possible to clone almost any identification (ID) tag used in present installations of the electronic access control systems (EACS), which use solely the IDs stored on tokens as the method to gain entry. Such tokens may be emulated by a suitable hardware, either by running a software on a micro-controller which mimics the identity token behavior, or by designing a copy of the identity token and intentionally removing the limitations enforced by the original identity token's manufacturer.

Originally, the emulation of identity tokens was limited to laboratory conditions [10] or specialized hardware [21,11] which required an experienced user and often hard-coding the ID in the source code. However, in recent years the alternatives to genuine tokens emerged, which are easy to use and do not require any prior user knowledge. We will concentrate on such devices and tokens, since they provide the easiest way to gain unauthorized access.

3.1 Dallas Semiconductors DS1990 iButton Serial Number Clone – RW1990

For quite some time, there was no official clone which would make it possible to emulate the iButton SN. However, during the last year, it became possible to order RW1990 chips encased in a steel case compatible with DS1990. The device comes with an empty ID, which must be programmed manually by cloning an existing DS1990 ID. Writing to the tags is done by a specialized 1-Wire protocol, for details see Figure 1.

On one hand, the commercial programmers for these tags are still expensive and can clone only the iButton SN which we have physical access to, on the other hand a generic solution can be implemented by uploading a publicly available sketch to Arduino and

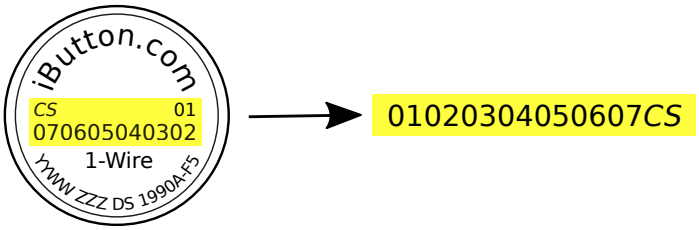

Fig. 2. iButton (DS1990) serial number extraction for programming of a RW1990 clone using the Arduino librarys (CS indicates the CRC8 checksum value)

use it to program the RW1990 chip. Moreover, since the ID of each iButton SN is laser-printed on its case, a simple photograph of the iButton is enough for its duplication by reversing the order of the bytes in the serial number (see Figure 2 for an example how the token ID may be read directly from a photo).

3.2 Passive Proximity EM41xx RFID Transponder Clone – EM4305

Fig. 3. Cloning of EM proximity 125 kHz tokens – from top-left corner clockwise: a USB writer, a hand-held copier, a blank card and a blank tag (with protective cover removed)

Tags used by so-called “card cloners” are based on E5550-compatible OTP identification circuit (Atmel ATA5557/5567/5577, etc.) These tags operate in 125 kHz range and provide bi-directional communication with the base station. Tags can be programmed

Fig. 4. Simplified ATA5577C [1] timing diagram during read and write operations. When the tag enters the RF field (1), the power-on reset circuit remains active until the internal capacitor is charged and an adequate voltage threshold is reached (2). After a default initialization delay (usually about 3 ms after entering the RF field), the tag enters regular read mode. In the normal mode, data is encoded according the configuration registers of ATA5577C and transmitted from the first stored data block (3). Most configurations use Manchester ASK modulation. To switch the tag from reading to writing mode, the card reader must interrupt the RF field for a short time (4), typically for 64 – 400 μs. After the successful mode switch, the tag enables (by default) the damping. A sequence of operational code, lock bits, address and data block must follow (5). Data bits are encoded by an on-off modulation with write gaps of 64 - 160 μs. The programming sequence is terminated with the programming delay of 5.6 ms. This delay is necessary for correct programming. After programming, the tag returns to block-read mode and transmits last written block (7).

to operate with different modulations and encodings, supporting also Manchester with ASK modulation used by the original EM4100 tags. The programming is performed by short interrupts in the RF field (on-off carrier keying) [1]. A simplified timing diagram of normal operation (reading mode) and programming mode is depicted in Figure 4.

3.3 Contactless RFID Tokens – UID Changeable MIFARE Classic Cards

In case of contactless RFID cards, there are several possible outcomes, which may or may not prevent successful cloning of the ID tag:

- The system uses ISO 14443 commands to read the tag ID and the ID is directly used for user access. In this case, a tag which allows writing of the UID may be used for cloning by a proximity attack.
- The system uses manufacturer-specific commands or utilizes the MIFARE classic cards and writes data to its memory. If no encryption is used, the proximity attack is possible, otherwise the attacker would need access to the ID token for several minutes to break the access keys first [10]. However, after that, a perfect one-to-one copy may be produced.
- The system uses manufacturer command set to read the ID token but its cryptography is broken – a clone supporting custom commands will be needed.
- Non-legacy card is being used – in such case a specialized equipment [18] is required to obtain the identity or duplication is not yet possible at all.

However, from our experience accessing the encrypted on-card memory with the reader significantly increases the processing time and requires more power harvested from the reader and better quality of the signal, otherwise the communication may

fail. As a result many manufacturers trade speed for security and use just the UID for obtaining access.

Examples of the tags used for cloning are Sector-0/Block-0 writable cards, emulating the common MIFARE Classic cards with 1kB of memory. These cards may be even written without any specialized hardware, using some common Android phones with NFC support or USB readers/writers. There are two types of such cards on market. The older ones are either directly writable, or they include specific commands which unlock the whole memory for reading and writing and bypass the security mechanisms. The more recent ones behave like standard MIFARE Classic cards, but allow to write the block in which the card ID is stored.

3.4 Overview of the Cloning Costs

The costs for cloning the ID tokens have presently become very low and the devices are easily available on popular auction sites with free delivery worldwide. The devices used to write data into cloned tags may be more expensive in some cases, but a cheap hardware may be used as an alternative and guides how to do that exist on Internet. In Table 1, we can see the overview of prices for such tags and programming devices.

Table 1. Approximate lowest prices of tags or cards (based on 5 pieces per lot) and devices used for identity token cloning – Q1/2017

Technology	Single tag/card	Standalone copier	USB writer	Custom built
DS1990 iButton	$ 1.4	$ 72 (TM1[3])	—	< $ 2
EM proximity 125 kHz	$ 0.5	$ 7 (ZX-6610)	$ 12	not needed
13.56 MHz RFID	$ 0.6 / $ 0.9	$ 50 [4]	$ 28 (ACR-122U)	< $ 7

4 Proposed Detection and Disabling Methods

In this section, we will propose the way how to detect or semi-permanently disable the cloned identity tokens and prevent the entry of unauthorized personnel. Generally, we will use the same techniques utilized for ID programming to detect such tokens. The hardware used for this task will be very cheap (less than $ 10 and for most cases less than $ 2), with the possibility to update the detection code and optional support for two different technologies (EM proximity 125 kHz tokens and iButtons) which should satisfy the manufacturers' requirements for minimal additional costs whilst improving the security.

The general mechanism is shown in Figure 5. Step one is typically necessary to detect the presence of an ID token, step two tries to use mechanism normally reserved

[3] It also supports cloning of EM proximity 125 kHz cards.

[4] We may use some smart-phones with NFC support which are still compatible with MIFARE Classic family instead.

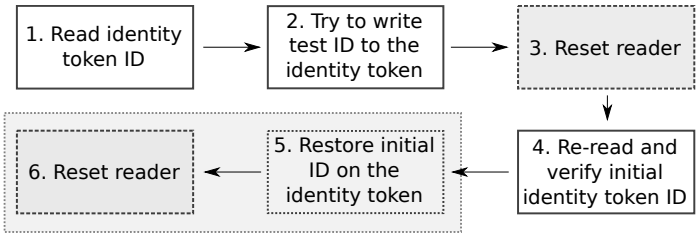

Fig. 5. Cloned token general detection method overview

for cloning (which original, read-only tokens will just ignore). Steps 3 and 6 may not be necessary in some implementations (we may write a gap instead in case of EM 125 kHz proximity cards). In step 4, the verification takes place and we check if the ID of the token has changed, indicating counterfeit token.

The fifth step is not compulsory for genuine tokens and may not be necessary for clones as well. On one hand, we may want to restore the original ID without letting the user know how the detection worked, on the other hand if this reader is used together with the original legacy readers, we may decide to semi-permanently disable the ID token and skip the restoration of the original ID altogether.

4.1 iButton RW1990 Clone

The detection follows steps 1–4 (or 1–6) shown in Figure 5 directly, the reader must be reset and chip initialized between reading and writing. Once the original iButton is detected, we reset the bus, change from the standard protocol to a lower-speed pulse-width modulation and send ID bytes to the tag, see Figure 1. Then we reset, switch back to the standard 1-Wire protocol and verify if the ID has changed. Optionally we may restore the original ID by step 5, resetting the bus before writing original ID again.

4.2 Passive Proximity RFID Transponder EM4305 Clone

The method used for detecting the most widespread clones of EM4100 tokens maintains the previous scheme. Proposed RFID reader (base station) reads the content of the tag memory and then tries to switch into the programming mode. As read only tags do not support this switch they will either continue in the reading sequence and resume sending the unique ID or will reset and start from the beginning if the write gap was long enough to trigger the reset. Both situations are easily recognizable. In case of programmable tag, the reader can rewrite (erase) the content of the tag and thus efficiently block it for later use or leave the content and just indicate that the tag is programmable and probably cloned. Figure 4 depicts such pattern. At first, normal read operation is performed, followed by a write gap, followed by a write phase.

Designs of 125 kHz RFID readers are quite simple. There are several off the shelf RFID reader OEM modules and integrated circuits (e.g. U2270B, HTRC110, IM283) implementing necessary RF circuits and signal processing. As the implementation itself

does not require any computational intensive tasks, it can be easily performed even on the smallest 8-bit microcontrollers, like Atmel AVRs. Thanks to this, it is possible to modify even existing designs by replacing the RFID controller with a microcontroller implementing proposed modification of the reading procedure.

4.3 UID Changeable MIFARE Classic Tags

For the MIFARE Classic clones, we have three different approaches how to detect the clone, however not all of them must be executed. In the first step, we read the token ID. Then we have three ways how to detect an unofficial ID token:

1. Issue the Request for Answer To Select (RATS) command and check the result. If the command is supported by the tag, a response is returned. We check the response for a special identification code (0xdabc1910) [19], if it is found, the token is an emulation token and we may ignore it (or write a special ID to it).
2. Issue special commands (7-bit value 0x40, then a single byte 0x43) [19]. If they are accepted, the token contains backdoor for unprotected access and we ignore the token by the reader (or write a special ID to it).
3. Try to write into first 16-byte block of the token (block-0/0), rewriting the original ID with a test one. After resetting the reader, we again check the ID of the token and if it has been changed to our test ID, we may either restore the token ID, or keep the test ID.

Tokens which were mentioned in the first and second cases were more expensive special emulation ones, which never became widespread due to their price. However, the first two detection methods may also be included, as they will prolong the test by approximately 150 ms in our test setup.

It should be noted that during the tests we have verified, that it is possible to semi-permanently disable the UID-writable tokens by writing zeros into the block-0/0, which will prevent tokens from being used by standard readers and from being detected by smart phones with NFC support (which can otherwise write to them). It is still possible to write a special software which recovers the token, but it is not easily available.

5 Experimental Evaluation of Our Approach

For the evaluation of our proposal for iButton and EM4100 clones, we have used a cheap hardware based on ATMega328p microcontroller, which is easily available for less than $ 2 due to its popularity in Arduino clones. It also provides serial output for potential readers. However, it is also possible to use cheaper processors such as ATtiny25/45/85.

The implementation of reading the 1-Wire iButton was straightforward as there are publicly available libraries implementing all necessary functions. The writing phase was implemented based on the Maxim's application note [13] and RW1990 descriptions available on Internet forums. Reading the token without the clone detection takes approximately 16 ms, while checking the clone by performing the write phase takes

additional 680 ms. The whole procedure of clone detection (as described in Section 4) takes approximately 712 ms. This is significantly longer time than the simple read, but still fast enough to not bother the end users.

Detecting EM4100 clones is much faster, mainly due to faster write operations. The normal reading is finished in about 33 ms and the writing phase takes approximately 90 ms. The cloned card can be detected even sooner, since the full write phase is not necessary. The decision can be made immediately after the reader generates the write gap (see Figure 4).

The most complex task was the evaluation of MIFARE Classic clones, which requires a specialized reader (e.g. with the PN532 chipset), already available in some devices. To simulate the solution we have written a tool using standard NFC libraries, executed on a PC with USB reader/writer.

In our test setup, it takes approximately 20 ms to select the token after initialization and obtain its UID (which would have to happen anyway). The actual detection methods from section 4.3 take approximately:

1. 25 ms for (unsuccessful) RATS query for our test tokens[5].
2. 65 ms for initial setup of the second test with special command sequence and
   - either 26 ms for successfully executed second test which identifies the token of this type.
   - or 63 ms for in case we use this test on and token incompatible with special commands (a genuine token or a token of the third type).
3. 30 ms for the test consisting of: authentication (8 ms), reading the block-0/0 backup (9 ms) and writing of the new block-0/0.

The reset step and following reinitialization of the token took 123 ms.

So even when we combine all of the abovementioned methods (in case of the genuine MIFARE Classic), the whole process still takes less than $\frac{1}{3}$ s, which is indistinguishable by the user. Whilst these times may be influenced by actual reader chipset and hardware setup, the total time should not be significantly higher.

To conclude the evaluation, all proposed clone detection mechanisms extend the ID tag processing times to some degree, but our observations have shown that the majority of users do not notice any change in reader behavior as it usually takes a few seconds to gain access using these technologies.

6 Conclusion

In this paper, we have shown some common, commercially available products used for identity token cloning. We have described the ways by which the cloning is achieved for three common legacy electronic access systems and presented a general framework and individual solutions for these systems, which would not significantly increase the time needed for a user to be verified by the system or the manufacturing costs, but which would provide an additional protection against cloned identity tokens. We have also discussed the time needed for our approach compared to normal verification time.

[5] We were not able to obtain a sample of an emulated card with these parameters for our testing.

In future, we may extend this detection framework by methods which check for other non-standard behavior of cloned ID tokens. This could, however, introduce situations, where counterfeit tags or generic tags from other manufacturers (which do not permit identity cloning) would be detected as false positives, preventing their use in such systems.

Acknowledgement This work has been supported in part by Grant of SGS No. SP2017/61, VŠB – Technical University of Ostrava, Czech Republic, and by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center – LM2015070".

References

1. Atmel Corporation: ATA5577C Read/Write LF RFID IDIC 100 to 150 khz (2014), technical Datasheet, rev. 9187H-RFID-07/14
2. Brandt, C.: Hacking iButtons. presentation at 27c3 (2010)
3. Brandt, C., Kasper, M.: Don't push it: Breaking iButton security. In: Foundations and Practice of Security, pp. 369–387. Springer (2014)
4. EM Microelectronic-Marin SA: EM4100 read only contactless identification device (2004), technical Datasheet
5. Garcia, F., de Koning Gans, G., Verdult, R., Meriac, M.: Dismantling iClass and iClass Elite. In: Foresti, S., Yung, M., Martinelli, F. (eds.) Computer Security - ESORICS 2012, Lecture Notes in Computer Science, vol. 7459, pp. 697–715. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-33167-1_40
6. Grand, J., Studio, G.I.: Can you really trust hardware? Exploring security problems in hardware devices. The Black Hat Briefings (2005)
7. Hancke, G.: A practical relay attack on ISO 14443 proximity cards. Tech. rep. (2005)
8. Hancke, G.: Practical attacks on proximity identification systems. In: Security and Privacy, 2006 IEEE Symposium on. pp. 6 pp.–333 (May 2006)
9. Issovits, W., Hutter, M.: Weaknesses of the ISO/IEC 14443 protocol regarding relay attacks. In: RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on. pp. 335–342 (Sept 2011)
10. de Koning Gans, G., Hoepman, J.H., Garcia, F.: A practical attack on the MIFARE Classic. In: Grimaud, G., Standaert, F.X. (eds.) Smart Card Research and Advanced Applications, Lecture Notes in Computer Science, vol. 5189, pp. 267–282. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-85893-5_20
11. Krumnikl, M.: Unique (EM4001) RFID emulator. Tech. rep., Department of Computer Science. VŠB - Technical University of Ostrava (2007)
12. Maxim Integrated Products, Inc.: DS1990A serial number iButton (2008), technical Datasheet
13. Maxim Integrated Products, Inc.: Software methods to achieve robust 1-Wire communication in iButton applications (2008), application Note 159
14. Mitrokotsa, A., Rieback, M., Tanenbaum, A.: Classifying RFID attacks and defenses. Information Systems Frontiers 12(5), 491–505 (2010), http://dx.doi.org/10.1007/s10796-009-9210-z
15. NXP Semiconductors: AN10927 MIFARE and handling of UIDs, application note, rev. 4.0

16. NXP Semiconductors: MF1S50yyX/V1 MIFARE Classic EV1 1K – Mainstream contactless smart card IC for fast and easy solution development, product data sheet, rev. 3.0
17. NXP Semiconductors: MF1S70yyX/V1 MIFARE Classic EV1 4K – Mainstream contactless smart card IC for fast and easy solution development, product data sheet, rev. 3.1
18. Oswald, D., Paar, C.: Breaking Mifare DESFire MF3ICD40: Power analysis and templates in the real world. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011, Lecture Notes in Computer Science, vol. 6917, pp. 207–222. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-23951-9_14
19. Tools, N.: Platform independent Near Field Communication (NFC) library. https://github.com/nfc-tools/libnfc (2017)
20. Verdult, R., Garcia, F.D., Balasch, J.: Gone in 360 seconds: Hijacking with Hitag2. In: Proceedings of the 21st USENIX Conference on Security Symposium. pp. 37–37. Security'12, USENIX Association, Berkeley, CA, USA (2012), http://dl.acm.org/citation.cfm?id=2362793.2362830
21. Verdult, R., de Koning Gans, G., Garcia, F.D.: A toolbox for RFID protocol analysis. In: RFID Technology (EURASIP RFID), 2012 Fourth International EURASIP Workshop on. pp. 27–34. IEEE (2012)