Mathematical and Computer Modelling 58 (2013) 219–226

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Legitimate-reader-only attack on MIFARE Classic

Ya Liu, Dawu Gu *, Bailan Li, Bo Qu

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Article history:
Received 18 February 2011
Received in revised form 10 July 2012
Accepted 24 July 2012

Keywords:
MIFARE Classic
CRYPTO1 cipher
Single authentication
Nested authentication
Proper tag nonces

ABSTRACT

MIFARE Classic is a contactless smart card which is widely used in several public transport systems. The researchers had presented different methods to clone a card in a practical card-only scenario. Among them, they recover the second or subsequent sector key by trying to accurately estimate the time information between two consecutive authentication attempts in a nested authentication. In this paper, we study the security of the MIFARE Classic in another practical scenario, where the adversary only communicates with a legitimate reader. The worst scenario to recover the second or subsequent sector key in a nested authentication only requires about 8 authentication attempts to the legitimate reader on average and the off-line search in about 328 s on Garcia's ordinary computer without estimating the time information between two consecutive authentications. Following this result, it is possible for the attackers to simulate or forge a legal card to authenticate successfully with a legitimate reader. To avoid this weakness, the reader must verify some information on the legal card at the beginning and it requires to be protected in some sense.

1. Introduction

The MIFARE Classic contact-less smart card is widely used in several public transport systems such as the Octopus card in Hong Kong, the Oyster card in London, OV-Chipkaart in Netherlands and so on. According to NXP, the MIFARE Classic covers more than 85% of the contactless smart card market. Recently, researchers have discovered that there exist serious flaws both from a cryptographic point of view and from a practical card-only scenario.

MIFARE Classic uses mutual authentication (a three-way protocol) and a proprietary filter stream cipher CRYPTO1 to guarantee data secrecy. In most applications, there are two kinds of authentications protocols, i.e., a signal authentication and a nested authentication. Generally, the nested authentication can provide more secure data protection compared with the signal authentication. In 2007, Nohl and Plötz partly reverse-engineered the hardware of the MIFARE Classic chip and recovered the algorithm CRYPTO1 by slicing the chip and taking pictures with a microscope [1]. In 2008, Gerhard de Koning Gans and Jaap-henk Hoepman studied the malleability of the stream cipher CRYPTO1 to read all memory blocks of the first sector of the card [2]. Then Dutch researchers mostly exploited the protocol vulnerabilities of MIFARE Classic [3], Nohl announced the first cryptographic attack on the system [4] after a report by Dutch security agency TNO that found the MIFARE Classic tags to be secure enough for some applications [5], and Courtois et al. examined MIFARE Classic from the aspect of algebraic attack [6]. Moreover, Garcia et al. also reverse-engineered MIFARE Classic based on the communication behavior between a tag and a reader [7]. They recovered the authentication protocol and the encryption algorithm, which unveiled several vulnerabilities. These results led to two attacks, one of which could derive the secret key in less than one second on ordinary hardware without pre-computation. However, the above attacks either require having access to a legitimate reader and a legal card or installing RF interception equipment inside a building. Although this is already

* Corresponding author.
E-mail addresses: liuyaloccs@gmail.com, liuya0611@sjtu.edu.cn (Y. Liu), dwgu@sjtu.edu.cn (D. Gu).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.07.020

disastrous from a cryptographic point of view, the application layer cannot allow these attacks to be performed undetected. In 2009, Garcia et al. and Courtois studied a more realistic card-only scenario, respectively. Specifically, Garcia et al. proposed four attacks [8], one of which could retrieve the key of the second authenticated sector in a nested authentication by estimating time information between two consecutive authentication attempts for every card, and Courtois presented an approach which recovered the key and cloned a card based on a simultaneous conditional differential property [9]. However, there are still two open problems that need to be studied further. First, the attack requires guessing many times in order to accurately estimate time information between two consecutive authentication attempts in a nested authentication. Therefore, is there any other method to recover the second or subsequent sector key in the nested authentication? Second, many readers are placed in public without any protection and the adversary can communicate with them easily. So, is the MIFARE Classic secure when the adversary only communicates with a legal reader?

In this paper, we study the security of MIFARE Classic in a legitimate-reader-only scenario for the first time, where an adversary only communicates with a genuine reader so as to recover the secret key. In a nested authentication, the online time to recover the sector key of the second or subsequent session is within negligible time, i.e., about 8 authentication attempts to the legitimate reader on average, and the off-line search takes under 328 s on Garcia's ordinary computer without estimating time information between two consecutive authentication attempts for every card. Following our results, the adversary can forge a legal card to communicate with a reader if that reader cannot authenticate any information on the tags. To avoid this weakness, the reader must verify some information on the legal card at the beginning and requires to be protected in some sense. Finally, we simulate our approach on an ordinary computer and list two examples when the key of the second or subsequent sector in a nested authentication can be recovered only by communicating with a legitimate reader.

The remainder of this paper is organized as follows. Section 2 briefly describes the logical structure of the MIFARE Classic, CRYPTO1 cipher, the authentication protocol, the parity bits and the generator of the tag nonce. Section 3 mainly presents how to retrieve the second or subsequent sector keys in a nested authentication only by communicating with a legitimate reader. Section 4 summarizes this paper.

2. Preliminaries

In this section, we briefly describe MIFARE Classic. The full description can be found in [8].

2.1. Communication layer and logical structure

The communication layer and data link layer of the MIFARE Classic card are based on the ISO 14443-A standard. However, MIFARE Classic adopts a proprietary protocol.

ProxMark is a generic device, which can be used to implement the functionalities of a card and a reader by using information from [3] about command codes of the MIFARE Classic and from [7] about its cryptographic aspects. Meanwhile, it can also eavesdrop on the communication between a reader and a tag including parity bits. Furthermore, it can be used to send any parity bits.

A MIFARE Classic card is in principle a memory chip with secure communication provisions. The memory is divided into some sectors, each of which can be further divided into blocks of 16 bytes each. The last data block of a sector is called the sector trailer, which stores two secret keys and the access condition for that sector. To perform an operation on a specific block, the reader must first authenticate for the sector including that block. The access conditions determine which key must be used.

2.2. CRYPTO1 cipher

The CRYPTO1 cipher is a filter stream cipher, which consists of a 48-bit LFSR with the feedback function $L : \mathbb{F}_2^{48} \to \mathbb{F}_2$ and a non-linear function $f : \mathbb{F}_2^{20} \to \mathbb{F}_2$, where

$$\begin{aligned} L(r_0, r_1, \ldots, r_{47}) \coloneqq\ & r_0 \oplus r_5 \oplus r_9 \oplus r_{10} \oplus r_{12} \oplus r_{14} \oplus r_{15} \oplus r_{17} \oplus r_{19} \oplus r_{24} \\ & \oplus r_{25} \oplus r_{27} \oplus r_{29} \oplus r_{35} \oplus r_{39} \oplus r_{41} \oplus r_{42} \oplus r_{43} \end{aligned}$$

and

$$\begin{aligned} f(r_0, r_1, \ldots, r_{47}) =\ & f_c(f_a(r_9, r_{11}, r_{13}, r_{15}), f_b(r_{17}, r_{19}, r_{21}, r_{23}), f_b(r_{25}, r_{27}, r_{29}, r_{31}), \\ & f_a(r_{33}, r_{35}, r_{37}, r_{39}), f_b(r_{41}, r_{43}, r_{45}, r_{47})), \end{aligned}$$

where f_a, f_b, f_c are defined in [8]. Let b_i be the keystream bits.

$$b_i = f(r_i, r_{i+1}, \ldots, r_{i+47}) = f(r_{i+9}, r_{i+11}, \ldots, r_{i+47}), \quad \forall i \geq 0.$$

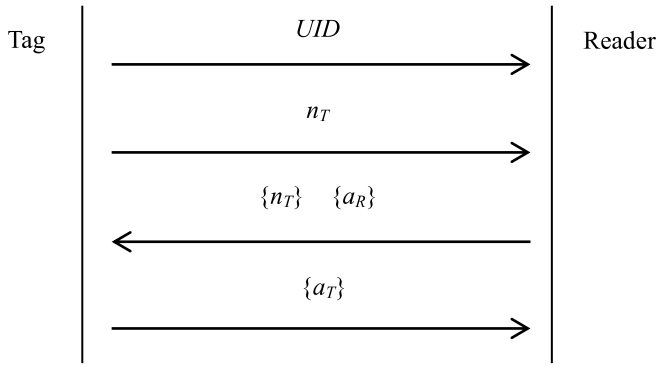

Fig. 1. A single authentication.

2.3. Authentication protocol

The authentication protocol was reverse-engineered in [7]. It makes use of a mutual three-pass authentication protocol. In most systems, there are two kinds of authentication protocol, i.e., a single authentication and a nested authentication.

2.3.1. A single authentication protocol and tag nonces

During the anti-collision phase, the tag sends its uid to the reader. The reader then asks to authenticate a request for a specific sector. Next, the tag picks a challenge nonce n_T. The reader sends back an 8-byte answer to that nonce which also contains a reader challenge of the reader n_R and the answer a_R. This answer is the first encrypted message after the authentication procedure. The tag finishes with its answer a_T. See the following Fig. 1.

In the authentication protocol, the 32-bit tag nonces are generated by a 16-bit LFSR with feedback polynomial $x^{16} + x^{14} + x^{13} + x^{11} + 1$. Every clock tick the LFSR shifts to the left and the feedback bit is computed using L_{16}, where $L_{16}(x_0x_1 \dots x_{15}) := x_0 \oplus x_2 \oplus x_3 \oplus x_5$.

Definition 2.1. A 32-bit value $n_T = n_{T,0}n_{T,1} \cdots n_{T,31}$ is a proper tag nonce if $n_{T,i} \oplus n_{T,i+2} \oplus n_{T,i+3} \oplus n_{T,i+5} \oplus n_{T,i+16} = 0$ for all $i \in \{0, 1, \dots, 15\}$. Denote the low-most 16 bits of n_T by $n_{T,0\sim15}$.

Definition 2.2 (*[8]*)**.** The successor function $\mathrm{suc} : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is defined by $\mathrm{suc}(x_0x_1 \dots x_{31}) := x_1x_2 \dots x_{31}L_{16}(x_{16}x_{17} \dots x_{31})$.

From [7], we know $a_R := \mathrm{suc}^{64}(n_T)$ and $a_T := \mathrm{suc}^{96}(n_T)$. Denote $a_{R,i} := \mathrm{suc}^{64}(n_T)_i$.

During the authentication protocol, the internal state of the stream cipher is initialized. In the following, we precisely define the initialization of the CRYPTO1 cipher and the generation of the LFSR-stream r_i for $i \in \mathbb{N}$.

Given a secret key $k = k_0k_1 \dots k_{47} \in \mathbb{F}_2^{48}$, a tag nonce $n_T = n_{T,0}n_{T,1} \dots n_{T,31} \in \mathbb{F}_2^{32}$, a card's ID $u = u_0u_1 \dots u_{31} \in \mathbb{F}_2^{32}$ and a reader nonce $n_R = n_{R,0}n_{R,1} \dots n_{R,31} \in \mathbb{F}_2^{32}$, the LFSR-state at time i is $\alpha_i := r_ir_{i+1} \dots r_{i+47} \in \mathbb{F}_2^{48}$, where $r_i \in \mathbb{F}_2$ are given by

$$r_i = \begin{cases} k_i, & i \in [0, 47]; \\ L(r_{i-48}, \dots, r_{i-1}) \oplus n_{T,i-48} \oplus u_{i-48}, & i \in [48, 79]; \\ L(r_{i-48}, \dots, r_{i-1}) \oplus n_{R,i-80}, & i \in [80, 111]; \\ L(r_{i-48}, \dots, r_{i-1}), & i \geq 112. \end{cases}$$

Furthermore, we denote encryptions by $\{-\}$ and define $\{n_{R,i}\}, \{a_{R,i}\}, \{a_{T,i}\} \in \mathbb{F}_2$ by

$$\{n_{R,i}\} := n_{R,i} \oplus b_{32+i}; \qquad \{a_{R,i}\} := a_{R,i} \oplus b_{64+i}; \qquad \{a_{T,i}\} := a_{T,i} \oplus b_{96+i}, \quad \forall i \in [0, 31].$$

2.3.2. A nested authentication protocol

Many systems authenticate for more than one sector. In a nested authentication, this is slightly different from the second authentication. At this time, the tag nonce is also sent encrypted with this sector key. Then the reader performs as before. In Fig. 2, we give the second or subsequent authentication session.

Definition 2.3 (*[8]*)**.** In a nested authentication, $\{n_T\} \in \mathbb{F}_2$ is defined by $\{n_{T,i}\} := n_{T,i} \oplus b_i$, $\forall i \in [0, 31]$. $\{n_R\}$ and $\{a_R\}$ are defined as before.

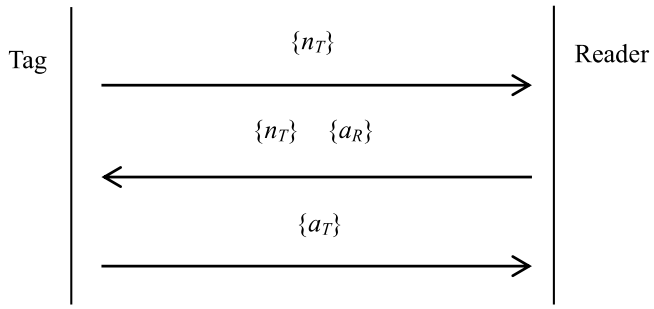

Fig. 2. The second or subsequent authentication in a nested authentication.

2.4. Parity

In ISO standard 14443-A [ISO01], every byte sent is followed by a parity bit. The MIFARE Classic computes parity bits over the plaintext instead of over the ciphertext. In addition, the parity bits are encrypted with the same keystream bit that is used to encrypt the next bit of plaintext. So every byte leaks one bit of information about the plaintext. The ISO standard specifies odd parity. We will give four parity bits of n_T, n_R, a_R and a_T, respectively.

Definition 2.4 ([8]). Define the parity bits $p_j \in \mathbb{F}_2$ by

$$p_j = \begin{cases} n_{T,8j} \oplus n_{T,8j+1} \oplus \cdots \oplus n_{T,8j+7} \oplus 1, & 0 \le j \le 3; \\ n_{R,8(j-4)} \oplus n_{R,8(j-4)+1} \oplus \cdots \oplus n_{R,8(j-4)+7} \oplus 1, & 4 \le j \le 7; \\ a_{R,8(j-8)} \oplus a_{R,8(j-8)+1} \oplus \cdots \oplus a_{R,8(j-8)+7} \oplus 1, & 8 \le j \le 11; \\ a_{T,8(j-12)} \oplus a_{T,8(j-12)+1} \oplus \cdots \oplus a_{T,8(j-12)+7} \oplus 1, & 12 \le j \le 15. \end{cases}$$

The encryptions $\{p_j\}$ are defined by $\{p_j\} \coloneqq p_j \oplus b_{8+8j}$, $\forall j \in [0, 11]$.

Theorem 1 ([8]). *For every $j \in 0, 1, 2$, we have $n_{T,8j} \oplus n_{T,8j+1} \oplus \cdots \oplus n_{T,8j+7} \oplus n_{T,8j+8} = \{p_j\} \oplus \{n_{T,8j+8}\} \oplus 1$.*

2.5. Rollback

Garcia et al. used the following rollback technique to recover the secret key because the inputs to the non-linear function don't include the first bit of the LFSR.

Definition 2.5 ([8]). Define a rollback function $R : \mathbb{F}_2^{48} \to \mathbb{F}_2$ by $R(r_1, r_2, \ldots, r_{48}) \coloneqq r_5 \oplus r_9 \oplus r_{10} \oplus r_{12} \oplus r_{14} \oplus r_{15} \oplus r_{17} \oplus r_{19} \oplus r_{24} \oplus r_{25} \oplus r_{27} \oplus r_{29} \oplus r_{35} \oplus r_{39} \oplus r_{41} \oplus r_{42} \oplus r_{43} \oplus r_{48}$.

Theorem 2 ([8]). *In a single authentication, we have*

$$r_i = \begin{cases} R(r_{1+i} \cdots r_{48+i}), & i \in [64, 95]; \\ R(r_{1+i} \cdots r_{48+i}) \oplus \{n_{R,i}\} \oplus f(0, r_{33+i}, \ldots, r_{79+i}), & i \in [32, 63]; \\ R(r_{1+i} \cdots r_{48+i}) \oplus n_{T,i} \oplus u_i, & i \in [0, 31]. \end{cases}$$

By Theorem 2, we can know any other internal state of the cipher α_i at any point i from a known LFSR internal state.

3. Main results

In this section, we study the security of the MIFARE Classic in a legal reader scenario for the first time, where the adversary only communicates with a reader. Specifically, we present a legitimate-reader-only attack on MIFARE Classic to recover the keys of all sectors according to two kinds of authentication, i.e., a single authentication and a nested authentication. In the whole attack, we require two assumptions. First, we assume that the attackers can obtain the card's ID by many approaches such as using ProxMark to eavesdrop a tag or guessing a card's ID by the rule of other card's ID number. Second, the attacker uses a ProxMark to simulate a card to communicate with a legal reader. In the following, we will introduce the attack procedure in detail.

In a single authentication, ProxMark first sends a random sequence as a challenge nonce of the tag n_T and the system doesn't verify a legal tag's information. Then the reader computes this sector's key by the tag's uid and a specific block requiring to be authenticated and sends its encrypted challenge nonce $\{n_R\}$ and the answer $\{a_R\}$. By intercepting two partial authentication sessions n_T, $\{n_R\}$, $\{a_R\}$, we can recover this secret key by using Garcia's second attack [8]. Up to now, if the system doesn't verify whether a card is legal or not, the adversary can derive the secret key of this sector.

In a nested authentication, we will show that the attacker may recover all authenticated sector keys in a legitimate-reader-only scenario without estimating any time information between two consecutive authentications. In the first authentication, the adversary uses the above method to recover the key of the first sector. For simplicity we assume that the first authentication can be verified successfully. We only study the subsequent authentication sessions. As a matter of fact, the ProxMark sends four bytes as the forged encryption tag nonce $\{n_T\}$ and four parity bits. The legitimate reader first checks the parity bits before answering the tag. If all four parity bits are correct, the reader will consider this random sequence as a correct tag nonce encrypted. Then it uses this sector's key to decrypt $\{n_T\}$ and obtain the low-most 16 bits of n_T, which can be used to initialize the 16-bit LFSR of the tag nonce. Finally, the reader runs the CRYPTO1 cipher, picks a challenge nonce n_R, computes $a_R = \mathrm{suc}^{64}(n_T)$ and sends eight bytes $\{n_R\}$ and $\{a_R\}$ to the tag. Clearly, the illegal tag doesn't respond at this moment.

It is important for the adversary to send the correct parity bits of a tag nonce. As a matter of fact, the attacker keeps the encrypted $\{n_T\}$ constant and has the freedom to send any parity bit. So the parity bits are correct with probability $\frac{1}{16}$. After on average 8 authentication sessions or at most 16, the reader can respond with $\{n_R\}$ and $\{a_R\}$ and the corresponding parity bits. Up to now, we have eavesdropped on the encryption $\{n_T\}$, $\{n_R\}$ and $\{a_R\}$. Thus, we can list the following equations:

$$\begin{aligned} r_{i+48} &= r_i \oplus r_{i+5} \oplus r_{i+9} \oplus r_{i+10} \oplus r_{i+12} \oplus r_{i+14} \oplus r_{i+15} \oplus r_{i+17} \oplus r_{i+19} \oplus r_{i+24} \oplus r_{i+25} \oplus r_{i+27} \\ &\quad \oplus r_{i+29} \oplus r_{i+35} \oplus r_{i+39} \oplus r_{i+41} \oplus r_{i+42} \oplus r_{i+43} \oplus n_{T,i} \oplus \mathrm{uid}_i, \quad i \in [0, 31] \end{aligned} \tag{1}$$

$$\begin{aligned} r_{i+80} &= r_{32+i} \oplus r_{i+37} \oplus r_{i+41} \oplus r_{i+42} \oplus r_{i+44} \oplus r_{i+46} \oplus r_{i+47} \oplus r_{i+49} \oplus r_{i+51} \oplus r_{i+56} \oplus r_{i+57} \\ &\quad \oplus r_{i+59} \oplus r_{i+61} \oplus r_{i+67} \oplus r_{i+71} \oplus r_{i+73} \oplus r_{i+74} \oplus r_{i+75} \oplus n_{R,i}, \quad i \in [0, 31] \end{aligned} \tag{2}$$

$$\begin{aligned} r_{i+112} &= r_{64+i} \oplus r_{i+69} \oplus r_{i+73} \oplus r_{i+74} \oplus r_{i+76} \oplus r_{i+78} \oplus r_{i+79} \oplus r_{i+81} \oplus r_{i+83} \oplus r_{i+88} \oplus r_{i+89} \oplus r_{i+91} \\ &\quad \oplus r_{i+93} \oplus r_{i+99} \oplus r_{i+103} \oplus r_{i+105} \oplus r_{i+106} \oplus r_{i+107}, \quad i \in [0, 31] \end{aligned} \tag{3}$$

$$\{n_{T,i}\} = n_{T,i} \oplus b_i = n_{T,i} \oplus f(r_i, r_{i+1}, \ldots, r_{i+47}) \quad i \in [0, 31] \tag{4}$$

$$\{n_{R,i}\} = n_{R,i} \oplus b_{i+32} = n_{R,i} \oplus f(r_{i+32}, r_{i+33}, \ldots, r_{i+79}) \quad i \in [0, 31] \tag{5}$$

$$\{a_{R,i}\} = a_{R,i} \oplus b_{i+64} = \mathrm{suc}^{64}(n_T)_i \oplus f(r_{i+64}, r_{i+65}, \ldots, r_{i+111}) \quad i \in [0, 31]. \tag{6}$$

Notation 3.1. *There are* 192 *equations with* 192 *variables in Eqs.* (1)–(6)*, which suggests one possible secret key. If the solution isn't unique, then the attacker will do this procedure again so as to recover an unique secret key by taking the intersection of the two sets of possible keys.*

However, we use ProxMark to emulate a card. Therefore, ProxMark sends a random 32-bit sequence as an encrypted tag nonce $\{n_T\}$ in the second authentication session. We can't guarantee that this random 32-bit sequence is obtained by encrypting a proper tag nonce n_T. (If $\{n_T\}$ is obtained by encrypting a proper tag nonce n_T, we call $\{n_T\}$ valid.) When $\{n_T\}$ is valid, we only need to solve Eqs. (1)–(6). When $\{n_T\}$ is invalid, the legal reader uses its secret key to decrypt the low-most 16 bits of $\{n_T\}$, which can be used to initialize LFSR. Using this 16-bit nonce, the reader generates a proper tag nonce n'_T. Clearly, $n_{T,0\sim15} = n'_{T,0\sim15}$ and $\{n_{T,0\sim15}\} = \{n'_{T,0\sim15}\}$. Using this proper tag nonce n'_T, the genuine reader runs its CRYPTO1 cipher, selects a challenge nonce n_R, computes an answer $a_R = \mathrm{suc}^{64}(n'_T)$ and sends eight bytes $\{n_R\}$ and $\{a_R\}$, which are generated by n'_T. So we can obtain some equations as follows:

$$\begin{aligned} r_{i+48} &= r_i \oplus r_{i+5} \oplus r_{i+9} \oplus r_{i+10} \oplus r_{i+12} \oplus r_{i+14} \oplus r_{i+15} \oplus r_{i+17} \oplus r_{i+19} \oplus r_{i+24} \oplus r_{i+25} \oplus r_{i+27} \\ &\quad \oplus r_{i+29} \oplus r_{i+35} \oplus r_{i+39} \oplus r_{i+41} \oplus r_{i+42} \oplus r_{i+43} \oplus n'_{T,i} \oplus \mathrm{uid}_i, \quad i \in [0, 31] \end{aligned} \tag{7}$$

$$\begin{aligned} r_{i+80} &= r_{32+i} \oplus r_{i+37} \oplus r_{i+41} \oplus r_{i+42} \oplus r_{i+44} \oplus r_{i+46} \oplus r_{i+47} \oplus r_{i+49} \oplus r_{i+51} \oplus r_{i+56} \oplus r_{i+57} \\ &\quad \oplus r_{i+59} \oplus r_{i+61} \oplus r_{i+67} \oplus r_{i+71} \oplus r_{i+73} \oplus r_{i+74} \oplus r_{i+75} \oplus n_{R,i}, \quad i \in [0, 31] \end{aligned} \tag{8}$$

$$\begin{aligned} r_{i+112} &= r_{64+i} \oplus r_{i+69} \oplus r_{i+73} \oplus r_{i+74} \oplus r_{i+76} \oplus r_{i+78} \oplus r_{i+79} \oplus r_{i+81} \\ &\quad \oplus r_{i+83} \oplus r_{i+88} \oplus r_{i+89} \oplus r_{i+91} \oplus r_{i+93} \oplus r_{i+99} \oplus r_{i+103} \oplus r_{i+105} \oplus r_{i+106} \oplus r_{i+107}, \quad i \in [0, 31] \end{aligned} \tag{9}$$

$$f(r_{i+9}, r_{i+11}, \ldots, r_{i+47}) = b_i = \{n'_{T,i}\} \oplus n'_{T,i}, \quad i \in [0, 15] \tag{10}$$

$$\{n_{R,i}\} = n_{R,i} \oplus b_{i+32} = n_{R,i} \oplus f(r_{i+32}, r_{i+33}, \ldots, r_{i+79}) \quad i \in [0, 31] \tag{11}$$

$$\{a_{R,i}\} = a_{R,i} \oplus b_{i+64} = \mathrm{suc}^{64}(n'_T)_i \oplus f(r_{i+64}, r_{i+65}, \ldots, r_{i+111}) \quad i \in [0, 31]. \tag{12}$$

Notation 3.2. *There are* 176 *equations with* 192 *variables in Eqs.* (7)–(12)*, which suggests about* $2^{192-176} = 2^{16}$ *possible solutions. It is difficult to solve a big system of multivariate equations with high degree. However, for the CRYPTO*1 *cipher, the tag uses a weak pseudorandom LFSR and the inputs to the non-linear function are only on odd-numbered places of the LFSR, which makes it possible to solve the above equations easily. The attacker will do this procedure again so as to obtain another* 2^{16} *possible solutions. By taking the intersection of the two sets, the adversary can retrieve the correct key in the second authentication.*

Notation 3.3. *In some applications, the legitimate reader may not deal with this as Notation 3.2. It may decrypt all 32 bits encrypted tag nonce and send these 32 bits tag nonce n_T to the CRYPTO1 cipher. The legal reader uses its high-most 16 bits to generate its answer a_R. For this scenario, the solving procedure is similar to the above case.*

In the following, we will present the whole procedure which can be used to solve Eqs. (7)–(12). This approach generalizes the methods of Section 6.3 of [8] and Section 4.4 of [9].

Procedure: We can present our ideas to solve these equations in the following. First, we try every possible tag nonce to recover 32 bits of keystream from Eq. (12), which will be used to retrieve all possible secret keys. Second, we verify whether these possible secret keys still satisfy other conditions. If some key satisfies other conditions, it will be a candidate for the correct key. We will give a full description below.

Step 1. *By parity bits of $\{n_T\}$ and $\{a_R\}$, we can reduce the number of all possible tag nonces from 2^{16} to 2^{12}.*

Since n'_T is a proper tag nonce, there are only 2^{16} possible challenge nonces of the tag. In general, we try to search all possible tag nonces n'_T to recover all key candidates. But the parity bits, which are sent with the encrypted tag nonce $\{n_T\}$ and an answer of the genuine reader $\{a_R\}$, leak some information about tag nonces. By Theorem 1, the parity bits sent with the encrypted tag nonce leak three bits of entropy. However we don't know if the reader checks the parity bits by n_T or n'_T. So at least a parity bit is correct by Theorem 1, i.e.,

$$\{p_0\} = n'_{T,0} \oplus n'_{T,1} \oplus \cdots \oplus n'_{T,7} \oplus n'_{T,8} \oplus \{n'_{T,8}\} \oplus 1. \tag{13}$$

On the other hand, the legitimate reader will send $\{n_R\}$ and $\{a_R\}$ with 8 parity bits to the tag. Thus, we obtain the following theorem:

Theorem 3. *The parity bits p_j defined as Definition 2.4. For every $j \in \{0, 1, 2\}$, we have $a_{R,8j} \oplus a_{R,8j+1} \oplus \cdots \oplus a_{R,8j+7} \oplus a_{R,8j+8} = \{p_{j+8}\} \oplus \{a_{R,8j+8}\} \oplus 1$.*

Proof. We can compute as follows by Definition 2.4:

$$\begin{aligned} a_{R,8j} \oplus a_{R,8j+1} \oplus \cdots \oplus a_{R,8j+7} \oplus a_{R,8j+8} &= p_{j+8} \oplus 1 \oplus a_{R,8j+8} \\ &= p_{j+8} \oplus b_{72+8j} \oplus a_{R,8j+8} \oplus b_{72+8j} \oplus 1 = \{p_{j+8}\} \oplus \{a_{R,8j+8}\} \oplus 1. \quad \square \end{aligned}$$

Lemma 3.1. *The parity bits p_j defined as Definition 2.4. We have*

$$\mathrm{suc}^{64}(n'_T)_{8j} \oplus \cdots \oplus \mathrm{suc}^{64}(n'_T)_{8j+8} = \{p_{j+8}\} \oplus \{\mathrm{suc}^{64}(n'_T)_{8j+8}\} \oplus 1, \quad j = 0, 1, 2. \tag{14}$$

Notation 3.4. *Eqs. (13) and (14) provide four bits of information on n'_T. So we can reduce the number of all possible tag nonces from 2^{16} to 2^{12}.*

Notation 3.5. *Eqs. (13) and (14) give four equations on n'_T. Adding to the above Eqs. (7)–(12), there are 180 equations with 192 variables, which suggests about $2^{192-180} = 2^{12} = 4096$ possible solutions.*

For each possible tag nonce n'_T, the adversary does Step 2.

Step 2. *For every possible tag nonce, the adversary can recover all possible secret keys. Moreover, a part of these possible keys which will be the candidate of the secret key will be found.*

First, for every possible tag nonce n'_T, the adversary can recover all key candidates. In fact, the attacker computes $a_R = \mathrm{suc}^{64}(n'_T)$ according to any possible tag nonce n'_T. From Eq. (12), he proceeds to calculate $b_{i+64} = \{a_{R,i}\} \oplus a_{R,i}$ for $i \in [0, 31]$, i.e.,

$$\begin{aligned} f(r_{64+i}, r_{65+i}, \ldots, r_{111+i}) &= f(r_{73+i}, r_{75+i}, \ldots, r_{111+i}) \\ &= b_{64+i} = \{a_{R,i}\} \oplus a_{R,i}, \quad i \in [0, 31]. \end{aligned}$$

Define odd tables T_i^O by

$$T_0^O := \{r_{73}r_{75}\cdots r_{111} \in \mathbb{F}_2^{20} | f(r_{73}, r_{75}, \ldots, r_{111}) = b_{64}\}$$

and for $1 \le i \le 15$

$$T_i^O := \{r_{73}r_{75}\cdots r_{111+2i} \in \mathbb{F}_2^{20+i} | r_{73}r_{75}\cdots r_{109+i} \in T_{i-1}^O \wedge f(r_{73+2i}r_{75+2i}\cdots r_{111+2i}) = b_{64+2i}\}.$$

Similarly, we define the even tables T_i^E by

$$T_0^E := \{r_{74}r_{76}\cdots r_{112} \in \mathbb{F}_2^{20} | f(r_{74}, r_{76}, \ldots, r_{112}) = b_{65}\}$$

and for $1 \le i \le 15$

$$T_i^E := \{r_{74}r_{76}\cdots r_{112+2i} \in \mathbb{F}_2^{20+i} | r_{74}r_{76}\cdots r_{110+i} \in T_{i-1}^E \wedge f(r_{74+2i}r_{76+2i}\cdots r_{112+2i}) = b_{65+2i}\}.$$

Table 1
Two examples.

	Example 1	Example 2
UID	0x00000000	0xC2A82DF4
$\{n_T\}$	0x11111111	0x4297C0A4
$\{p_0p_1p_2p_3\}$	0111	1011
Auth 1: $\{n_R\}$, $\{a_R\}$	0x5B7490359DB00F5B	0x3AA0D7EDB7A55A99
$\{p_4p_5p_6p_7p_8p_9p_{10}p_{11}\}$	11100100	10101111
N	4068	4143
Auth 2: $\{n'_R\}$, $\{a'_R\}$	0x1BA7D0F7656D581B	0xFA1B70BEFF85AB98
$\{p'_4p'_5p'_6p'_7p'_8p'_9p'_{10}p'_{11}\}$	01001000	01101001
KEY	0x00FF00FF00FF	0xD3EE426948D8

Since f is balanced, $|T_0^O| = |T_0^E| = 2^{19}$. An entry in T_i^O can be obtained by adding a matching $r_{112+2i} \in \{0, 1\}$ at the end of an entry T_{i-1}^O for $1 \le i \le 15$. On average, $\frac{1}{2}$ of the time the entry in T_{i-1}^O can be extended only with 0 or only with 1, $\frac{1}{4}$ of the time the entry can be extended with 0 and with 1, and $\frac{1}{4}$ of the time the entry can be deleted. Consequently, $|T_i^O| \approx 2^{19}$. Similarly, $|T_i^E| \approx 2^{19}$.

In the following, we will verify whether the combination of two elements separately from two tables T_{15}^O and T_{15}^E satisfies the LFSR feedback function. To reduce the search complexity, we split the feedback into an even part and an odd part by their contribution of the feedback polynomial. We first define the odd part of the feedback function, $L^O : \mathbb{F}_2^{24} \to \mathbb{F}_2$, by $L^O(x_1x_3 \cdots x_{47}) := x_5 \oplus x_9 \oplus x_{15} \oplus x_{19} \oplus x_{25} \oplus x_{27} \oplus x_{29} \oplus x_{35} \oplus x_{39} \oplus x_{41} \oplus x_{43}$ and the even part of the feedback function, $L^E : \mathbb{F}_2^{24} \to \mathbb{F}_2$, by $L^E(x_0x_2 \cdots x_{46}) := x_0 \oplus x_{10} \oplus x_{12} \oplus x_{14} \oplus x_{24} \oplus x_{42}$. Clearly, $L(x_0x_1x_2 \cdots x_{47}) = L^E(x_0x_2 \cdots x_{46}) \oplus L^O(x_1x_2 \cdots x_{47})$. Then we denote the contribution of the entries of the odd table to the feedback, $\psi^O : T^O \to \mathbb{F}_2^{22}$, by

$$\psi^O(r_{73}r_{75} \cdots r_{143}) := (L^E(r_{73+2i} \cdots r_{119+2i}), L^O(r_{75+i}, \ldots, r_{121+2i}))_{i\in[0,10]}$$

and define the contribution of the entries of the even table to the feedback, $\psi^E : T^E \to \mathbb{F}_2^{22}$, by

$$\psi^E(r_{74}r_{76} \cdots r_{142}) := (L^E(r_{74+2i} \cdots r_{120+2i}), L^O(r_{76+i}, \ldots, r_{122+2i}))_{i\in[0,10]}.$$

Denote the combined table T^C as

$$T^C := \{r_{73}r_{74}r_{75} \cdots r_{142} \in \mathbb{F}_2^{70} | r_{73}r_{75} \cdots r_{141} \in T_{15}^O \wedge r_{74}r_{76} \cdots r_{142} \in T_{15}^E \wedge \psi^O(r_{73}r_{75} \cdots r_{141}) = \psi^E(r_{74}r_{76} \cdots r_{142})\}.$$

If the sequence $r_{73}r_{74} \cdots r_{142}$ is indeed generated by the LFSR, then $r_{73}r_{74} \cdots r_{142} \in T^C$.

Up to now, the attacker has retrieved a table T^C including all possible LFSR-states $r_{73}r_{74} \cdots r_{142}$. By Theorem 2, we first retrieve all possible LFSR-states $r_{32}r_{33} \cdots r_{142}$ before the tag nonce is fed in. Then we recover the secret key $r_0r_1 \cdots r_{47}$ by Eq. (7).

Second, the attacker will find some of the possible keys which will be the candidate for the secret key. Given a tag nonce, the adversary can recover all possible secret keys. Then for this tag nonce, the attackers will list those keys which produce the correct encrypted tag nonce $\{n'_{T,0-15}\}$. If it generates the correct low-most 16 bits, it will be a candidate for the secret key. Otherwise, it's not.

In conclusion, the adversary attempts 8 authentications on average and the off-line search takes $2^{12} * 0.04 \approx 164\ s$ to recover approximately 2^{12} possible solutions on Garcia's ordinary computer, if he can implement it with the hardware configuration used in [7]. In other words, the attacker obtains a set containing possible candidates of the secret key. □

Repeating the above procedure again, we can obtain another set that consists of all possible key candidates. By taking the intersection of these two sets consisting of key candidates, the attacker can retrieve the correct key. During the whole attack procedure, the attacker requires 8 authentications on average and spends 328 s to derive a correct secret key on Garcia's ordinary computer.

We simulate a nested authentication and recover the second authenticated sector's key. In Table 1, we list two examples where we can recover the second sector's key in a nested authentication by eavesdropping on two series of partial authentication sessions and corresponding parity bits. Before this, we give some notation. For example, uid is a card's ID, $\{n_T\}$ is the encrypted tag nonce, $\{p_0p_1p_2p_3\}$ are the correct encrypted parity bits sent with $\{n_T\}$, $\{n_R\}$ and $\{a_R\}$ are the answer of the legitimate reader in an authentication session and the corresponding encrypted parity bits are $\{p_4p_5p_6p_7p_8p_9p_{10}p_{11}\}$. In another authentication session, the genuine reader responds $\{n'_R\}$ and $\{a'_R\}$ with correct encrypted parity bits $\{p'_4p'_5p'_6p'_7p'_8p'_9p'_{10}p'_{11}\}$. N is the number of all possible solutions using authentication session one.

To summarize, if the reader does not verify the card's information, the adversary can recover the keys of all authenticated sectors by only communicating with a genuine reader no matter whether this sector takes part in a single authentication or a nested authentication. This results in some serious vulnerabilities so that an adversary may forge an legal card if they can satisfy one of two conditions. First, the adversary can guess a legal card's ID. Applying the above methods, the attacker may obtain the keys of all authenticated sectors. Second, he may counterfeit an illegal card whose ID doesn't belong to this system, provided the system does not verify whether the card is legal or not. By communicating with a legitimate reader, the attacker can obtain all sectors' keys corresponding to this ID and write some information according to another legal card. Then this illegal card may communicate with a genuine reader.

4. Conclusion

In this paper, we have proposed a legitimate-reader-only attack on MIFARE Classic for the first time, where an adversary only communicates with a genuine reader. We find that the key of the second or subsequent sectors in a nested authentication can be recovered just by wireless eavesdropping on the communication between a genuine reader and an emulator, ProxMark. Specifically, the attackers only attempt 8 authentications on average, negligible online time, and the off-line search takes under 328 s to recover the key of the second sector on Garcia's ordinary computer if we can implement it with the hardware configuration used in [2]. Following these results, some tags may be stolen or some illegal tag will be counterfeited without rhyme or reason. To avoid this weakness, the reader must be protected in some sense and the reader is required to verify some information on the tag. Finally, we simulate our attacks and list two examples where the key of the second authenticated sector can be recovered in a nested authentication.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (No. 61073150) and development and industrial applications of a series of secure and independent cryptographic smart card chips.

References

[1] K. Nohl, H. Plötz, Mifare-little security despite obscurity, Presentation on the 24th Congress of the Chaos Computer Club, Berlin, 2007. http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html.
[2] G. de koning Gans, J.H. Hoepman, F.D. Garcia, A practical attack on the MIFARE Classic, in: CARDIS 2008, in: LNCS, vol. 5189, Springer-Verlag, Heidelberg, 2008, pp. 267–282.
[3] R.W. Schreur, P.V. Rossum, F.D. Garcia, W. Teepe, J. Hoepman, B. Jacobs, G. Gans, R. Verdult, R. Muijrers, R. Kali, V. Kali, Security flaw in MIFARE Classic [EB/OL], Press release, Digital Security Group, Radboud University Nijmegen, 2008. http://www.sos.cs.ru.nl/rfid/pressrelease.en.html.
[4] K. Nohl, Cryptanalysis of CRYPTO-1 [EB/OL]. http://www.cs.virginia.edu/kn5f/Mifare.Cryptanalysis.html.
[5] TNO, Security analysis of the dutch OV-Chipkaart [EB/OL]. http://www.tno.nl.
[6] N.T. Courtois, K. Nohl, S. O'Neil, Algebraic attacks on the CRYPTO-1 stream cipher in MIFARE Classic and Oyster cards [EB/OL]. http://eprint.iacr.org/2008/166.pdf.
[7] F.D. Garcial, G. Gans, R. Muijrers, P. Rossum, R. Verdult, R.W. Schreur, B. Jacobs, Dismantling MIFARE Classic, in: ESORICS 2008, in: LNCS, vol. 5283, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 97–114.
[8] F.D. Garcia, P.V. Rossum, R. Verdult, R.W. Schreur, Wirelessly pickpocketing a MIFARE Classic card [C], in: IEEE Symposium on Security and Privacy 2009, pp. 3–15, 2009.
[9] N.T. Courtois, The dark side security by obscurity and cloning MIFARE Classic rail and building passes [EB/OL]. http://eprint.iacr.org/2009/137.pdf.