Brute Force Cryptanalysis of MIFARE Classic Cards on GPU

Cihangir Tezcan
Department of Mathematics, Middle East Technical University, Ankara, Turkey
Informatics Institute, Department of Cyber Security, CYDES Laboratory, Middle East Technical University, Ankara, Turkey
cihangir@metu.edu.tr

Keywords: MIFARE Classic, CRYPTO1, Cryptanalysis, GPU.

Abstract: MIFARE Classic is the most widely deployed contactless smartcard on the market. However, many active and passive attacks are provided after its proprietary stream cipher CRYPTO1 was reverse engineered. The short 48-bit key of the CRYPTO1 cipher, leaked parity bits and the encrypted error code that is sent after a failed authentication (which is corrected in the hardened new cards) allow the adversary to perform offline brute force attack and avoid detection. Such an attack requires wireless interaction with a card for less than a second and then a brute force attack which was shown to take around 9 days on a single GTX280 GPU. We optimized this brute force attack on modern GPUs by using bitsliced implementation technique and observed that a brute force attack on a GTX970 GPU can be performed in less than 5 hours. Although this attack is not applicable to hardened MIFARE Classic cards, a similar attack using the short key length and the leaked parity bits can be performed when a single key is known, possibly using the default keys for unused sectors. Such an attack requires wireless interaction with a card for less than a second and then a brute force attack which was shown to take approximately one month on a single GTX460 GPU. Our bitsliced implementation of this attack takes less than 7 hours on a GTX970 GPU.

1 INTRODUCTION

ISO/IEC 14443-A standard about identification, contactless integrated circuit, and proximity cards consists of four parts: Physical characteristics, radio frequency power and signal interface, initialization and anticollision, and transmission protocol. MIFARE Classic, which is the most widely used contactless smartcard on the market, is compatible with the first three parts but it uses its own secure communication layer. It uses a proprietary stream cipher called CRYPTO1 to provide data confidentiality and authentication between card and reader. Although kept secret by the manufacturer NXP Semiconductors, both the cipher CRYPTO1 (Nohl et al., 2008) and the communication layer (Garcia et al., 2008) have been reverse engineered.

After the reverse engineering, many vulnerabilities and attacks were provided. Most of the card-only attacks rely on non-cryptographically related implementation flaws and some of these flaws are mitigated by issuing replacement cards.

Currently the best known active card-only attack is the ciphertext-only attack of (Meijer and Verdult, 2015), which only requires wireless interaction with the card for a few minutes with consumer-grade hardware. Active attacks on these cards are summarized in Table 1.

Moreover, offline attacks on these cards are possible due to the short key and the parity bit vulnerability. If the card also has the keystream leakage through error vulnerability, it was shown in (Chih et al., 2010) that the brute force attack provided in (Garcia et al., 2009) can be performed in around 9 days on a single GTX280 GPU. If the card does not have this vulnerability, it was shown in (Meijer and Verdult, 2015) that a brute force attack using the nested authentication property can be performed in a month using a single GTX460 GPU provided that a single key of a sector is already known. Thus, one can perform these attacks in a few hours by building a huge farm of GPUs. However, we observed that the CRYPTO1 stream cipher of the MIFARE Classic card is very suitable for a bitsliced (Biham, 1997) optimization and we reduced the time of these attacks to 5 and 7 hours, respectively on a single GTX970 GPU.

Tezcan, C.
Brute Force Cryptanalysis of MIFARE Classic Cards on GPU.
DOI: 10.5220/0006262705240528
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 524-528
ISBN: 978-989-758-209-7

Table 1: Comparison of card only attacks.

Attack	Traces	Gather	Compute	[a]	[b]
(Garcia et al., 2009)	2	<1 sec	<1 sec	×	✓
(Courtois, 2009)	300	3 min	<1 sec	×	×
(Chiu et al., 2013)	$\sim 100,000$	10-20 hours	2-15 min	✓	×
(Meijer and Verdult, 2015)	$\sim 10,000$	6-12 min	5-10 min	✓	✓

[a]Does not require a weak PRNG
[b]Does not require the error code after a failed authentication

2 MIFARE CLASSIC CARDS

2.1 Memory Structure

The memory of a MIFARE Classic card is divided into sectors that are divided into 16-byte blocks. The last block of each sector stores two sector keys and the access conditions for that sector. To perform an action on a block, the reader must first authenticate itself for that sector with a sector key.

2.2 CRYPTO1

Nohl et al. reverse engineered CRYPTO1 stream cipher by slicing a MIFARE Classic chip and taking pictures with a microscope (Nohl et al., 2008). It consists of a 48-bit linear feedback shift register (LFSR) and a non linear filter function f. Contents of the LFSR are shifted one position to the left and the most significant bit is discarded. A new bit is generated by the feedback function L. During the authentication phase, the input is also XORed to the output of L.

Definition 2.1. *(Nohl et al., 2008) The feedback function $L(x_0x_1\dots x_{47}) : \mathbb{F}_2^{48} \to \mathbb{F}_2$ is defined by*

$$\begin{aligned} L := \quad & x_0 \oplus x_5 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus \\ & x_{19} \oplus x_{24} \oplus x_{25} \oplus x_{27} \oplus x_{29} \oplus x_{35} \oplus x_{39} \oplus x_{41} \\ & \oplus x_{42} \oplus x_{43} \end{aligned}$$

Definition 2.2. *(Garcia et al., 2008) The filter function $f(x_0x_1\dots x_{47}) : \mathbb{F}_2^{48} \to \mathbb{F}_2$ is defined by*

$$\begin{aligned} f := f_c(\ & f_a(x_9,x_{11},x_{13},x_{15}), f_b(x_{17},x_{19},x_{21},x_{23}), \\ & f_b(x_{25},x_{27},x_{29},x_{31}), f_a(x_{33},x_{35},x_{37},x_{39}), \\ & f_b(x_{41},x_{43},x_{45},x_{47}) \ ) \end{aligned}$$

and $f_a, f_b : \mathbb{F}_2^4 \to \mathbb{F}_2$ and $f_c : \mathbb{F}_2^5 \to \mathbb{F}_2$ are defined by

$$\begin{aligned} f_a(y_0,y_1,y_2,y_3) := \ & ((y_0 \vee y_1) \oplus (y_0 \wedge y_3)) \oplus \\ & (y_2 \wedge ((y_0 \oplus y_1) \vee y_3)) \\ f_b(y_0,y_1,y_2,y_3) := \ & ((y_0 \wedge y_1) \vee y_2) \oplus \\ & ((y_0 \oplus y_1) \wedge (y_2 \vee y_3)) \\ f_c(y_0,y_1,y_2,y_3,y_4) := \ & (y_0 \vee ((y_1 \vee y_4) \wedge (y_3 \oplus y_4))) \\ & \oplus ((y_0 \oplus (y_1 \wedge y_3)) \wedge \\ & ((y_2 \oplus y_3) \vee (y_1 \wedge y_4))) \end{aligned}$$

If we denote the LFSR-stream with $a_0a_1\dots$ and keystream $b_0b_1\dots$, they are obtained as follows:

Definition 2.3. *(Garcia et al., 2009) Given a 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 uid $u = u_0u_1\dots u_{31} \in \mathbb{F}_2^{32}$, and a reader nonce $n_R = n_{R,0}n_{R,1}\dots n_{R,31} \in \mathbb{F}_2^{32}$, the internal state of the cipher at time i is $\alpha_i := a_ia_{i+1}\dots a_{i+47} \in \mathbb{F}_2^{48}$.*

Here the $a_i \in \mathbb{F}_2$ are given by

$$\begin{aligned} a_i := \quad & k_i & & \forall i \in [0,47] \\ a_{48+i} := \quad & L(a_i,\dots,a_{47+i}) \oplus n_{T,i} \oplus u_i & & \forall i \in [0,31] \\ a_{80+i} := \quad & L(a_{32+i},\dots,a_{79+i}) \oplus n_{R,i} & & \forall i \in [0,31] \\ a_{112+i} := \quad & L(a_{64+i},\dots,a_{111+i}) & & \forall i \in \mathbb{N} \end{aligned}$$

Furthermore, the keystream bit $b_i \in \mathbb{F}_2$ at time i is defined by

$$b_i := \quad f(a_ia_{1+i}\dots a_{47+i}) \quad \forall i \in \mathbb{N}$$

Structure of CRYPTO1 stream cipher is provided in Figure 1.

2.3 Tag and Reader Authentication Protocol

The reverse engineered authentication protocol (de Koning Gans et al., 2008) is as follows: The tag is selected in the anticollision phase and it sends its uid u to the reader. Then the reader asks to authenticate for a specific memory block b. Consequently the tag sends a challenge nonce n_T. The reader responds by encrypting its own challenge n_R and the answer $a_R = suc^{64}(n_T)$ where

$$suc(x_0x_1\dots x_{31}) := x_1x_2\dots x_{31}L_{16}(x_{16}x_{17}\dots x_{31})$$

and

$$L_{16}(x_0x_1\dots x_{15}) := x_0 \oplus x_2 \oplus x_3 \oplus x_5.$$

The authentication is concluded with the tag answer $a_T = suc^{96}(n_R)$.

2.4 Known Vulnerabilities

Previous works provided many serious vulnerabilities of MIFARE Classic. They are thoroughly explained in (Meijer and Verdult, 2015) and we summarize the ones that are used in this paper:

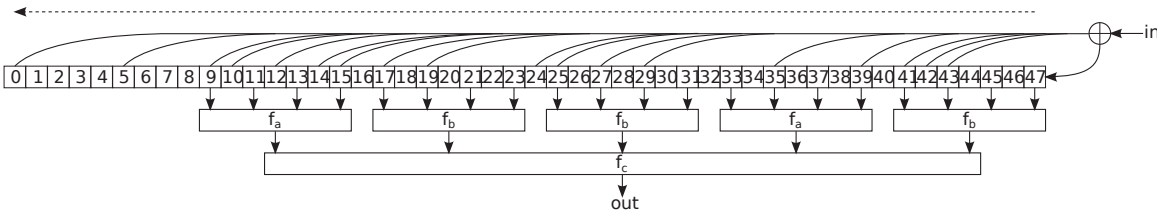

Figure 1: Structure of CRYPTO1 stream cipher.

1. Short key length: The key size of 48 bits is too small. Although the delay introduced by the communication and authentication procedure prevents an online brute force attack, reverse engineering of CRYPTO1 algorithm allowed offline brute force attacks.
2. The nested authentication: When the authentication for a sector is completed, the reader can request another authentication for a different sector and key. This request initializes the internal cipher state to the key of that sector. But this time, the nonce n_T is also sent encrypted. In case the card has the weak pseudo-random number generator vulnerability, the attack provided in (Garcia et al., 2009) can be used to recover 32 bits of keystream by only wirelessly interacting with a card.
3. Parity bits: The MIFARE Classic sends a parity bit for each byte it transmits but these parity bits are computed over plaintext instead of ciphertext.
4. Keystream leakage through error: During authentication protocol, the card always first checks the parity bits and if the parity bits are correct but the answer a_R is wrong, the card provides the $0x5$ failed authentication error code. But this error is sent encrypted and thus 4 bits of the keystream is leaked. This weakness is removed with hardened MIFARE Classic cards since they do not send authentication error code.

Note that it is important for the adversary to retrieve all data for every sector because deployed systems using MIFARE Classic sequentially authenticate for several sectors verifying the data in the card. It was observed in (Meijer and Verdult, 2015) that the only way left in hardened cards that leak secret key information without communicating with a genuine reader is through parity bits. Using this vulnerability, (Meijer and Verdult, 2015) provided an online attack for these hardened cards that requires data gathering for 6-12 minutes and an offline attack that requires wireless interaction with a card for less than a second and then a brute force attack which take approximately one month on a single GTX460 GPU. Authors noted that their implementation is not bitsliced (Biham, 1997) and a bitsliced implementation would improve the attack performance by at least a factor of four. In Section 3.2, we show that a bitsliced optimization of this attack provides much better speed ups and the attack takes only 7 hours on a GTX970 GPU.

3 BRUTE FORCE ATTACKS ON GPU

CRYPTO1 stream cipher uses a short key of 48 bits, tag serial, tag nonce, and reader nonce to produce keystream. Thus, if the attacker captures the keystream and the nonces, then a 48-bit exhaustive search on a GPU can find the key in less than an hour. In this attack the adversary can check the keystream for each bit produced by a key and thus early abort the search. However, the proposed attacks in the literature cannot capture the keystream directly but can capture the parity of bytes of the keystream. Moreover, early versions of MIFARE classic cards send 4-bit encrypted error code after failed authentication. Thus, we can divide brute force attacks on MIFARE Classic into two categories which depends on if the card sends encrypted error code after failed authentication or not.

3.1 Brute Force using Encrypted Error Code

The brute force attack in the presence of encrypted error code is proposed in (Garcia et al., 2009). In this attack, the attacker tries to authenticate for a sector and answers the challenge of the tag with eight random bytes and eight random parity bits for n_R and a_R. When the parity bits are correct but the answer a_R is wrong, the tag sends the encrypted 4-bit error code. Thus, a success leaks 12 bits of entropy about the key. Therefore, we need at least 4 authentication sessions with correct parity bits but in practice 6 authentication sessions are generally used to avoid false positives. Since the probability of correctly guessing 8 parity bits is $\frac{1}{256}$, gathering six authentication sessions with correct parity bits takes $6 \cdot 256 = 1536$ au-

thentication attempts on average and it can be done in less than one second.

In (Garcia et al., 2009), it was pessimistically assumed that COPACOBANA which finds a 56-bit DES key in on average 6.4 days can find the 48-bit CRYPTO1 key in 36 minutes if once can fit the same number of CRYTO1 checks on FPGA as DES-decryptions. Note that COPACOBANA costs approximately 10,000 USD. In another work (Chih et al., 2010), it was shown that 16 GTX280 GPUs can perform this attack in 14 hours. Thus, it takes around 9.3 days for a single GTX280 (240 cores, processor clock 1296 MHz) to perform this attack.

In our bitsliced implementation, we kept the 48-bit keys using 48 32-bit registers and thus performed each operation on 32 different keys simultaneously. We also kept the nonce, uid, and parity information on the shared memory of the GPU to reduce the number of used registers. Data on the shared memory can be reached on 32 memory lanes and two threads in warp trying to use the same warp causes collision. However, if each 32 threads in a warp tries to access the same memory bank, then the data is broadcast and no collision occurs. Since every thread uses the same nonce, uid, and parity information, no collisions occur in our implementation. We used CUDA SDK version 6.5 instead of 7.5 because it provided better results. Our implementation uses 154 registers, 2080 bytes of shared memory and 360 bytes of constant memory when compiled with compute capability 5.0 or 5.2 but number of registers and performance vary for different compute capability. We noted that using 48 64-bit registers for bitslicing increases the number of registers and reduce the performance. We performed our experiments on a medium range desktop and a laptop GPUs. Brute force attack results are provided in Table 2.

Table 2: Brute Force Attack on MIFARE Classic.

	GTX 860M	GTX 970
Cores	640	1664
Clock	1020 MHz	1253 MHz
Compute Capability	5.0	5.2
Keys per second	6,673 M	15,575 M
48-bit search	11.7 hours	5 hours

3.2 Brute Force Attack on Hardened MIFARE Classic

It was noted in (Meijer and Verdult, 2015) that most deployed systems leave default keys intact for unused sectors and nearly all deployed systems that use key diversification leave at least one sector key non-diversified because it is recommended in the manufacturer guidelines for system integrators. Therefore, we can assume that the adversary always knows at least one key of a sector. Once the adversary authenticates against that sector, they can send another authentication request for a different sector and key. This new authentication command sets the internal state of the cipher to the key of the new sector but this time the challenge generated by the tag n_T is also sent encrypted. Thus, the adversary can perform brute force attack on the parity bits of the encrypted nonce sent by the tag. Since an encrypted 32-bit nonce has 4 parity bits, on average 12 encrypted nonces are required to determine the 48-bit key. This requires wireless interaction with a card for less than a second.

A brute force attack with an Nvidia GTX460 GPU (336 cores, 1350 MHz processor clock) was performed in (Meijer and Verdult, 2015) and it was deduced that a full 48-bit exhaustive search would take approximately 1 month. Thus, the authors conclude that 180 Nvidia GTX460 GPUs would cost around $\$12,600$ USD and would recover a single key within an hour. However, this implementation is not bitsliced and CRYPTO1 stream cipher is very suitable for bitsliced implementation on GPUs.

In our bitsliced implementation, we kept the 48-bit keys using 48 32-bit registers and thus performed each operation on 32 different keys simultaneously. We also kept the nonce and parity information on the shared memory of the GPU to reduce the number of used registers. Since every thread uses the same nonce and parity information, no collisions occur in our implementation when shared memory is accessed. We again used CUDA SDK version 6.5 instead of 7.5 because it provided better results. Our implementation uses 95 registers, 1952 bytes of shared memory and 356 bytes of constant memory when compiled with compute capability 5.0 or 5.2 but number of registers and performance vary for different compute capability. We noted that using 48 64-bit registers for bitslicing increases the number of registers and reduce the performance. We performed our experiments on a medium range desktop and a laptop GPU. Brute force attack results are provided in Table 3.

Table 3: Brute Force Attack on Hardened MIFARE Classic.

	GTX 860M	GTX 970
Cores	640	1664
Clock	1020 MHz	1253 MHz
Compute Capability	5.0	5.2
Keys per second	3,635 M	11,105 M
48-bit search	21 hours	7 hours

This attack is slower than the brute force attack of Section 3.1 because in this case we do not have the leaked 4-bit keystream. This is because hardened MIFARE classic cards do not send encrypted er-

ror code. However, we get a speed up because our implementation uses only 95 registers per thread of the GPU, compared to 154 registers of the attack of Section 3.1. This way we can call 512 threads per block of GPU to get better occupancy, compared to 256 threads used in the other attack. During these experiments, although our GTX970 GPU has a processor clock of 1253 MHz, it was clocked to 1329 MHz due to its GPU Boost technology.

4 CONCLUSIONS

MIFARE Classic is the most widely deployed contactless smartcard on the market and many vulnerabilities are provided in the literature. Offline attacks on these cards are possible due to the short key, parity bit vulnerability, and keystream leakage through error vulnerability. In this work, we optimized these brute force attacks on GPUs using a bitsliced implementation and observed that it takes less than only 5 hours to perform the attack of (Garcia et al., 2009) on a single GTX970 GPU when we have the keystream leakage through error vulnerability. Hardened cards do not have this vulnerability and we showed that the brute force attack of (Meijer and Verdult, 2015) for these cards takes around 7 hours to perform it on a single GTX970 GPU. Thus, we show that these brute force attacks to clone MIFARE Classic cards are way more practical than it was assumed.

ACKNOWLEDGEMENTS

This work was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant 115E447 titled "Quasi-Differential Factors and Time Complexity of Block Cipher Attacks".

REFERENCES

Biham, E. (1997). A fast new DES implementation in software. In Biham, E., editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 260–272. Springer.

Chih, M.-Y., Shih, J.-R., Yang, B.-Y., Ding, J., and Cheng, C.-M. (2010). MIFARE Classic: Practical attacks and defenses. In *Proceedings of the 19th Cryptology and Information Security Conference (CISC 2010)*, Hsinchu, Taiwan.

Chiu, Y., Hong, W., Chou, L., Ding, J., Yang, B., and Cheng, C. (2013). A practical attack on patched MIFARE classic. In Lin, D., Xu, S., and Yung, M., editors, *Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27-30, 2013, Revised Selected Papers*, volume 8567 of *Lecture Notes in Computer Science*, pages 150–164. Springer.

Courtois, N. (2009). The dark side of security by obscurity - and cloning mifare classic rail and building passes, anywhere, anytime. In Fernández-Medina, E., Malek, M., and Hernando, J., editors, *SECRYPT 2009, Proceedings of the International Conference on Security and Cryptography, Milan, Italy, July 7-10, 2009, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*, pages 331–338. INSTICC Press.

de Koning Gans, G., Hoepman, J., and Garcia, F. D. (2008). A practical attack on the MIFARE classic. In Grimaud, G. and Standaert, F., editors, *Smart Card Research and Advanced Applications, 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings*, volume 5189 of *Lecture Notes in Computer Science*, pages 267–282. Springer.

Garcia, F. D., de Koning Gans, G., Muijrers, R., van Rossum, P., Verdult, R., Schreur, R. W., and Jacobs, B. (2008). Dismantling MIFARE classic. In Jajodia, S. and López, J., editors, *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer.

Garcia, F. D., van Rossum, P., Verdult, R., and Schreur, R. W. (2009). Wirelessly pickpocketing a mifare classic card. In *30th IEEE Symposium on Security and Privacy (S&P 2009), 17-20 May 2009, Oakland, California, USA*, pages 3–15. IEEE Computer Society.

Meijer, C. and Verdult, R. (2015). Ciphertext-only cryptanalysis on hardened mifare classic cards. In Ray, I., Li, N., and Kruegel, C., editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 18–30. ACM.

Nohl, K., Evans, D., Starbug, and Plötz, H. (2008). Reverse-engineering a cryptographic RFID tag. In van Oorschot, P. C., editor, *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*, pages 185–194. USENIX Association.