Cryptanalytic Attacks on Cyber-Physical Systems

Martin Novotný
Czech Technical University in Prague
Faculty of Information Technology
Department of Digital Design
novotnym@fit.cvut.cz

Abstract Cryptography finds its application in various objects used in our everyday life. GSM communication, credit cards, tickets for public transport or RFID tags employ cryptographic features either to protect privacy or to ensure trustworthy authentication. However, many such objects are vulnerable to certain cryptanalytic attacks. In this review we discuss how FPGA-based cryptanalytic hardware may compromise GSM communication, or how standard laboratory equipment may be used for breaking Smart Card security. This review summarizes keynote speech that was given at 5th Mediterranean Conference on Embedded Computing (MECO'2016).

Keywords Cryptography, Cryptanalysis, Attacks, Cyber-Physical Systems, GSM, Car Immobilizers, Door Opening Systems, Public Transport Tickets, Differential Power Analysis

1 Introduction

Cryptography had been prevalently and almost solely used by government agencies and defense administrations for strategic purposes until about middle of 20th century. Since 1970's, with the dawn of electronic communication and growing production of consumer electronics based on digital technology, cryptography becomes more and more important in everyday lives of all of us. Telephone conversations over GSM network are encrypted to protect our privacy. Communication between credit card and the ATM or card reader is encrypted as well. Car immobilizers and tags opening the car doors use cryptography to provide trustworthy authentication and to prevent fraudulent duplication of such tags. RFID cards and tags used in door opening

systems are equipped with cryptographic features to ensure unique identification of the card/tag holder. We find cryptography also in the tickets for public transport or in modern RFID tags used for identification of goods. In many of these commonly used devices we employ cryptography for two main purposes — either to protect our privacy (e.g. in GSM network), or to ensure trustworthy authentication (e.g. in car immobilizers or door opening systems).

Unfortunately, many cryptographic systems built in modern devices can be easily compromised. This is primarily for three main reasons:

- Short keys are used or the cryptographic system has a low information entropy. For example, many car immobilizers use Hitag-2 cipher [1] with 48-bit key, while the keys shall be at least 80 bits long.
- No randomness or almost no randomness is employed. For example, obsolete Mifare cards with Crypto-1 cipher used the same data during every challenge-response protocol [2]. It was easy to record such communication and replay it later. Another example is Keeloq cipher, used in car opening systems. This cipher works with device key, which is derived from known serial number of the device [3].
- Cryptographic system is vulnerable to side-channel attack. For example, observing the power consumption [4] of the cryptographic device, or measuring the time needed for cryptographic operation may reveal the secret key used during encryption.

In this review we discuss several examples of frequently used objects, whose cryptographic features have been severely threaten. In section 3 we discuss security of GSM communication. In sections 4 and 5 we present attacks breaking the car security. While in section 4 we present attacks on Hitag-2 cipher used in car immobilizers, in section 5 we introduce attacks on KeeLoq cipher used in car door opening systems. Section 6 is dedicated to public transport tickets equipped with Mifare DESfire chip. In the last section 7 we conclude this review.

2 Attacker's Equipment

Equipment used for breaking the ciphers depends on the type of attack the cryptanalyst (or hacker) plans to mount on the cryptographic system.

Brute-force attacks, guess-and-determine attacks, or precomputation of large tables for time-memory trade-off attacks typically require high computational effort, while neither high-speed communication nor large memory is

Figure 1: Photo of COPACOBANA. Courtesy of SciEngines GmbH

demanded. As these tasks are parallelizable onto independent nodes — all nodes are running the same algorithm, but each node is processing its unique set of data — one can use e.g. a cluster of supercomputers. However, it is more efficient to employ cluster of FPGAs, as FPGA-based systems provide better price/performance, as well as better energy/performance ratio [5]. For example, expenses for building cluster COPACOBANA [5, 6] (see photo at Figure 1) were about € 10,000 and the cluster is able to reveal the DES key in less than two weeks. Cluster of computers (composed of then PCs) breaking the DES key at the same time would have to contain more than 32,000 units [5, 7]. Expenses for building cluster of computers would be about 650 times higher, and the energy consumption of such cluster would be about 8,000 times higher than the energy consumption of COPACOBANA. Therefore, FPGA-based systems like COPACOBANA (equipped with 120 Xilinx Spartan-3 1000 FPGAs) or its younger sister RIVYERA [8] (equipped with up to 128 Xilinx Spartan-6 LX150 FPGAs) have been frequently used for breaking ciphers like DES [6], A5/1 [5, 9], Hitag-2 [10] or KeeLoq [11].

When having a physical access to a cryptographic device, the attacker can use differential power analysis (DPA) [3, 4] or some other type of side-channel attack. In such case the attacker typically needs an oscilloscope, which is a standard equipment of lab, and additional proprietary kits that in many cases may be purchased for expenses well below € 100. Currently, DPA and similar attacks belong to the most powerful methods, while being

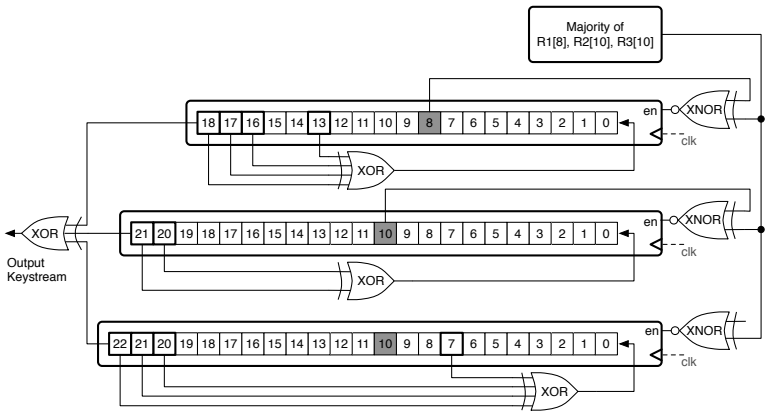

Figure 2: A5/1 cipher

relatively cheap for breaking cryptographic systems.

3 Breaking GSM Communication

Communication between a base transceiver station (BTS) and a mobile phone is encrypted with A5/1 cipher. A5/1 is a stream cipher — data representing a telephone call (plaintext) are mixed (xored) with the pseudo-random keystream produced by A5/1 cipher and the result (ciphertext) is then transmitted between mobile phone and BTS. The internal structure of A5/1 cipher is depicted in Figure 2. It consists of three linear feedback shift registers (LFSR) that are irregularly clocked. Most significant bits of LFSRs are mixed together to produce the output keystream. The internal state, i.e. the content of all three registers, changes between clock cycles. If the internal state at certain point of encryption is revealed, then the cipher is broken, as it is possible to clock the cipher forward, as well as to backtrack the cipher to previous states.

The attacker may intercept encrypted communication between mobile phone and BTS. As some known data are also encrypted, it is possible, by mixing known plaintext with intercepted ciphertext, to obtain the keystream. Therefore, many proposed attacks [12, 13, 14, 15, 16, 17] are focused on deriving the internal state that produced the keystream (now known). These attack proposals are realistic due to relatively low entropy of A5/1 cipher — internal state has just 64 bits and some proposed attacks may have even lower complexity. In this review we present two real-world implementations

Table 1: Number of guesses that need to be checked for consistency.

Type of the attack	Attack complexity
Plain brute-force attack	2^{64}
Plain guess-and-determine attack	$2^{41} \times 2^{11} = 2^{52}$
Smart guess-and-determine attack [9]	$2^{41} \times (2 - \frac{1}{4})^{11} \approx 2^{50}$

of attacks on A5/1 that employ FPGA-based machine COPACOBANA.

3.1 Guess-and-Determine Attack on A5/1

While plain brute-force attack would require to check 2^{64} guesses, there are more efficient approaches. For example, it is possible to guess the content of registers $R1$ (19 bits), $R2$ (22 bits) and lower 11 bits of register $R3$. After that the cipher is clocked and upper bits of $R3$ are derived from the known keystream and the content of $R1$ and $R2$. If any contradiction appears, the guess is discarded and the next guess is verified. Such guess-and-determine attack would need to check only $2^{19+22+11} = 2^{52}$ guesses.

In [9] we present the improved guess-and-determine attack that we implemented in COPACOBANA. In this case we guess the content of registers $R1$ and $R2$. Lower 11 bits are gradually guessed during clocking. If we face contradiction on the guessed value of bit, we stop generating the remaining lower bits of $R3$. Due to this fact, we do not check all 2^{11} combinations of lower 11 bits of $R3$, but only $(2 - \frac{1}{4})^{11}$ such combinations, leading to overall complexity of $2^{41} \times (2 - \frac{1}{4})^{11} \approx 2^{50}$ guesses. Attack complexities are summarized in Table 1.

With COPACOBANA, we can check all 2^{50} guesses within 11.78 hours, in other words, an average time to reveal the internal state is just 5.89 hours.

3.2 Time-Memory Trade-off Attack on A5/1

In 1981, Martin Hellman proposed the idea of time-memory trade-off (TMTO) tables attack [18]. He demonstrated this attack on block cipher DES. The attack consists of two phases — offline (precomputation) phase and online (attack) phase. In offline phase, large tables for given cipher are precomputed. Data are organized in chains, out of which only start points and end point are stored, thus reducing memory complexity. Data contained in these tables are later used during online phase to significantly speedup the attack. Offline phase has large complexity, comparable to brute-force attack, however, it is run only once, while generated tables may be used repeatedly

Table 2: A5/1 TMDTO: Expected runtimes and memory requirements

#	m	S	d	I_l	PT [days]	M [TB]	T [secs]	TA	P_{total}
1	2^{41}	2^{15}	5	$[2^3,2^6]$	337.5	7.49	70.5	2^{21}	0.86
2	2^{39}	2^{15}	5	$[2^3,2^7]$	95.4	3.25	92.0	2^{21}	0.67
3	2^{40}	2^{14}	5	$[2^4,2^7]$	95.4	4.85	27.6	2^{20}	0.63
4	2^{40}	2^{14}	5	$[2^3,2^6]$	84.4	7.04	17.7	2^{20}	0.60
5	2^{39}	2^{15}	5	$[2^3,2^6]$	84.4	3.48	70.5	2^{21}	0.60
6	2^{40}	2^{14}	5	$[2^4,2^6]$	84.4	5.06	21.5	2^{20}	0.55
7	2^{37}	2^{15}	6	$[2^4,2^8]$	47.7	0.79	186.3	2^{21}	0.42
8	2^{36}	2^{16}	6	$[2^4,2^8]$	47.7	0.39	745.3	2^{22}	0.42

during online phase(s). Note that the success ratio strongly depends on the amount of generated data. As the data are generated in pseudo-random manner, the success ratio never reaches 100%.

Several modifications of original Hellman's idea exist. Ron Rivest proposed using so-called distinguished points (DP) [19] to accelerate search in tables during online phase. Biryukov and Shamir introduced the time-memory-data trade-off (TMDTO) approach applicable to stream ciphers [20]. Oechslin proposed so-called rainbow tables [21] and Barkan, Biham and Shamir later proposed its modification called thick-rainbow and thin-rainbow tables [22].

In our attack implemented in COPACOBANA we combined TMDTO approach with distinguished points. We experimented with various parameters setups, like the number of generated chains m, the length of one thin-rainbow sequence S and the DP-property d, as shown in Table 2. For example, if we focus on row 3 of Table 2, then for the set of parameters $m = 2^{40}$, $S = 2^{14}$ and $d = 5$, the precomputation of tables would take $PT = 95.4$ days and the precomputed data would occupy $M = 4.85$ terabytes of disk space. In the online phase, after completing all the calculations in just $T = 27.6$ seconds, and after performing $T = 2^{20}$ table accesses, the cipher may be broken with the probability of $P_{total} = 63\%$.

If the result is not found, then the attack described in subsection 3.1 may be applied.

4 Stealing the Car I — Car Immobilizers with Hitag-2 Cipher

Hitag-2 is a stream cipher that is allegedly used in car immobilizers of producers like BMW, Audi, Alfa Romeo, Porsche, Bentley, VW, Peugeot, Renault, Citroën, Iveco trucks, etc. [23]. The cipher suffers from very short key being only 48 bits long. Therefore, after reverse-engineering of the cipher [24], the same authors proposed the algebraic attack [25]. Their attack requires 4 sniffed communications between a RFID transponder built in the car key, and a read-write device that is built in the ignition barrel inside the car. Hitag-2 key is revealed after 45 hours of calculations on a standard PC.

Due to the short key the brute-force attack is also easily applicable when implemented in hardware. To demonstrate this, we have implemented brute-force attack in COPACOBANA [10]. Our attack needs just 2 sniffed communications between the transponder and the read-write device. The key is revealed by COPACOBANA after just 103.5 minutes at maximum. This attack also demonstrates the power of FPGA-based systems, as already mentioned in Chapter 2 — software implementation of brute-force attack would need 4 years to complete all calculations on a standard PC. In this case, COPACOBANA equipped with 120 Xilinx Spartan-3 1000 FPGAs has the power comparable to more than 20,000 PCs.

Team from Radboud University, Netherlands and KU Leuven, Belgium later published even faster attack [23]. To improve the efficiency of the attack, they utilize several vulnerabilities they have found in Hitag-2 cipher. For the attack they need to sniff 136 communications between the transponder and the read-write device. Following calculations need just 5 minutes on quadcore laptop. The authors claim that TMTO attack based on their attack would require just 1 minute.

Attacks mentioned above also demonstrate the diversity of complexities of various types of attacks. While plain brute-force attack, simply checking all 2^{48} combinations of the key until the match is found, would need 4 years of computational time on a single PC (or less than 2 hours on COPACOBANA), algebraic attack [25] would require 45 hours. The fastest reported attack, utilizing the weaknesses of the cipher as much as possible [23], would need just 5 minutes. On the other hand, the brute force attack needs data from just 2 sniffed communications that may be obtained remotely e.g. by sensitive antenna (and without necessity of physical access to the device), while fastest attack needs data from 136 communications. As the communication between the transponder and the read-write device

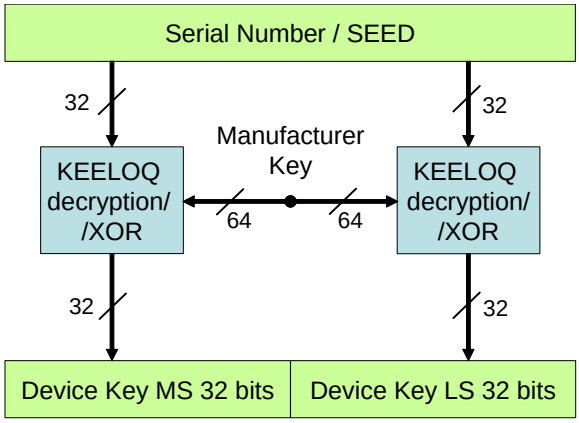

Figure 3: KeeLoq device key generation.

is initiated only upon ignition, collecting the necessary amount of data via remote sniffing may last a long time. In this case, only possession of a car key with built-in transponder would make the attack practical.

5 Stealing the Car II — KeeLoq & Opening the Car Doors

KeeLoq is allegedly used in tags for the remote opening of the car doors of producers like Chrysler, Daewoo, Fiat, GM, Honda, Toyota, Volvo, Volkswagen Group, Clifford, Shurlok, Jaguar, etc. The cipher was broken by team of Ruhr-University in Bochum [3]. By means of differential power analysis they are able to reveal both the device key stored in a tag and the manufacturer key stored in car reader. While the device key is unique for each device (tag), the manufacturer key is shared among all car readers of the same producer. This represents significant threat, as the manufacturer key is used for generation of device key, as shown in Figure 3.

The device key is generated from the tag serial number, which is transmitted in plain between the tag and the car. Therefore, if the attacker knows the manufacturer key for a given type of car, he just needs to sniff 1-2 communications between the tag and the car and to compute corresponding device key.

Lower 32 or 48 or 60 bits of the serial number can be replaced by randomly generated seed, nevertheless, many car producers do not utilize this property. However, even using 32 or 48 randomly generated bits would not

Table 3: Worst case times for the brute force attack on KeeLoq

SEED length (bits)	1 FPGA (<€ 80)	1 COPACOBANA (<€ 10,000)	100 COPACOBANAs (<€ 1,000,000)
32	39 secs	0.33 secs	3.3 msecs
48	29.6 days	5.9 hours	213 secs
60	332 years	1011 days	10.1 days

secure the system too much, as we can employ COPACOBANA or another FPGA-based system again. Figures for COPACOBANA-based attack [11] are presented in Table 3.

Researchers from Tel Aviv University [26] further improved COPACOBANA-based attack by utilizing special properties of Xilinx FPGAs, where some Look-Up Tables (LUTs) can be configured as shift registers (denoted as SRL16). They modified attack architecture to employ SRL16, which led to more dense design running at higher frequency. The attack could be completed in one third of the original time then. Authors have also implemented their attack in the newer families of Xilinx FPGAs. For revealing the device key derived from 48 bit seed they need either 17 hours with single Virtex-4 FPGA, or just 3 hours with single Virtex-6 FPGA.

6 Public Transport for Free

Many contactless cards used e.g. as subscription tickets for public transport use some of MIFARE chips from NXP Semiconductors N.V. For example, Oyster Cards in London were equipped with MIFARE Classic chips using Crypto-1 cipher. After being reverse-engineered the cipher was quickly broken [2]. As a result, all new Oyster Cards issued after December 2009 use MIFARE DESFire EV1 chips [27].

London is not the only capital in the world where traveling in public transport for free is (or was) possible. For example, OpenCards, issued in Prague, Czech Republic, use the MIFARE technology as well. Older cards are based on MIFARE DESFire technology equipped with chip MF3ICD40, while newer cards are based on MIFARE DESfire EV1 technology.

Older version of MIFARE DESFire (chip MF3ICD40) was broken by researchers from Ruhr-University Bochum [28]. They used the differential power analysis again. To break the chip they needed to measure 250,000 power traces, which took about 7 hours. Overall expenses for the attack did not exceed € 2,000. The researchers are able to read all the files stored in

this older type of OpenCard, to read the master key (which is identical for all the OpenCards!) and to read 3 special keys.

Naturally, the producer of MIFARE technology (NXP Semiconductors N.V.) employed state-of-the-art countermeasures against side-channel attacks into their newer products. Until now, there is no side-channel attack on either MIFARE DESFire EV1 or MIFARE DESFire EV2 reported in an open literature.

7 Conclusions

This review was just a brief enumeration of several attacks that were mounted on object we are frequently using in our everyday lives. List of examples of cyber-physical systems that are vulnerable to some type of attack may continue.

Presented attacks on Hitag-2, KeeLoq and Crypto-1 prove that so-called "security by obscurity" does not in fact represent any security at all. The ciphers were broken soon once their internal structure was revealed. Therefore, cryptographic systems shall be based on ciphers and principles that were thoroughly examined and discussed within cryptographic community.

References

[1] N. T. Courtois, S. O'Neil, and J.-J. Quisquater, "Practical Algebraic Attacks on the Hitag2 Stream Cipher," in *ISC '09: Proceedings of the 12th International Conference on Information Security*, vol. 5735 of *LNCS*, pp. 167–176, Springer, 2009.

[2] K. Nohl, D. Evans, S. Starbug, and H. Plötz, "Reverse-engineering a Cryptographic RFID Tag," in *Proceedings of the 17th Conference on Security Symposium*, SS'08, (Berkeley, CA, USA), pp. 185–193, USENIX Association, 2008.

[3] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme," in *Advances in Cryptology — CRYPTO 2008*, pp. 203–220, 2008.

[4] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology — CRYPTO' 99*, pp. 388–397, Springer Berlin Heidelberg, 1999.

[5] T. Güneysu, T. Kasper, M. Novotný, C. Paar, and A. Rupp, "Cryptanalysis with COPACOBANA," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 57, no. 11, pp. 1498–1513, 2008.

[6] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, "Breaking Ciphers with COPACOBANA — A Cost-Optimized Parallel Code Breaker," in *Proceedings of CHES'06*, vol. 4249 of *LNCS*, pp. 101–118, Springer-Verlag, 2006.

[7] M. Novotný, "Al Gore and NSA: A Match Made in Heaven Thanks to COPACOBANA. Talk given at the Rump Session of CHES 2008," August 12 (2008).

[8] T. Güneysu, T. Kasper, M. Novotný, C. Paar, L. Wienbrandt, and R. Zimmermann, "High-Performance Cryptanalysis on RIVYERA and COPACOBANA Computing Systems," in *High Performance Computing Using FPGAs*, pp. 335–366, Springer-Verlag, 2013.

[9] T. Gendrullis, M. Novotný, and A. Rupp, "A Real-World Attack Breaking A5/1 within Hours," in *Proceedings of the 10th Workshop on Cryptographic Hardware and Embedded Systems (CHES 2008)*, pp. 266–282, Springer-Verlag, 2008.

[10] P. Štembera and M. Novotný, "Breaking Hitag2 with Reconfigurable Hardware," in *Proceedings of the 14th Euromicro Conference on Digital System Design*, (Los Alamitos), pp. 558–563, IEEE Computer Society Press, 2011.

[11] M. Novotný and T. Kasper, "Cryptanalysis of KeeLoq with COPACOBANA," in *Workshop on Special Purpose Hardware for Attacking Cryptographic Systems (SHARCS 2009)*, pp. 159–164, 2009.

[12] R. Anderson, "A5 (Was: HACKING DIGITAL PHONES)." sci.crypt, 17 June 1994.

[13] J. Golic, "Cryptanalysis of Alleged A5 Stream Cipher," in *Proc. of Eurocrypt'97*, vol. 1233 of *LNCS*, pp. 239–255, Springer-Verlag, 1997.

[14] E. Biham and O. Dunkelman, "Cryptanalysis of the A5/1 GSM Stream Cipher," in *Proc. of Indocrypt'00*, vol. 1977 of *LNCS*, Springer-Verlag, 2000.

[15] A. Biryukov, A. Shamir, and D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC," in *Proc. of FSE'00*, vol. 1978 of *LNCS*, pp. 1–18, Springer-Verlag, 2001.

[16] P. Ekdahl and T. Johansson, "Another Attack on A5/1," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 284–289, 2003.

[17] J. Keller and B. Seitz, "A Hardware-Based Attack on the A5/1 Stream Cipher," 2001.

[18] M. E. Hellman, "A Cryptanalytic Time-Memory Trade-Off," *IEEE Transactions on Information Theory*, vol. 26, pp. 401–406, 1980.

[19] D. E. R. Denning, *Cryptography and Data Security*. Addison-Wesley, 1982.

[20] A. Biryukov and A. Shamir, "Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers," in *Proc. of Asiacrypt'00*, vol. 1976 of *LNCS*, pp. 1–13, Springer, 2000.

[21] P. Oechslin, "Making a Faster Cryptanalytic Time-Memory Trade-Off," in *Proc. of CRYPTO'03*, vol. 2729 of *LNCS*, pp. 617–630, Springer, 2003.

[22] E. Barkan, E. Biham, and A. Shamir, "Rigorous Bounds on Cryptanalytic Time/Memory Tradeoffs," in *Proc. of CRYPTO'06*, vol. 4117 of *LNCS*, pp. 1–21, Springer, 2006.

[23] R. Verdult, F. D. Garcia, and J. Balasch, "Gone in 360 seconds: Hijacking with Hitag2," in *21st USENIX Security Symposium (USENIX Security 2012)*, pp. 237–252, USENIX Association, 2012.

[24] N. Courtois and S. O'Neil, "Reverse-Engineered Philips/NXP Hitag2 Cipher. Talk given at the Rump Session of Fast Sotware Encryption conference (FSE 2008)," February 12 (2008).

[25] N. T. Courtois, S. O'Neil, and J.-J. Quisquater, "Practical Algebraic Attacks on the Hitag2 Stream Cipher," in *Information Security: 12th International Conference, ISC 2009, Pisa, Italy, September 7-9, 2009. Proceedings*, pp. 167–176, Springer Berlin Heidelberg, 2009.

[26] I. Sheetrit and A. Wool, "Cryptanalysis of KeeLoq Code-Hopping Using a Single FPGA," 2011.

[27] Wikipedia, "Oyster card — Wikipedia, The Free Encyclopedia," 2017. [Online; accessed 20-January-2017].

[28] D. Oswald and C. Paar, "Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World," in *Cryptographic Hardware and Embedded Systems – CHES 2011. Proceedings*, pp. 207–222, Springer Berlin Heidelberg, 2011.