Smart Card Fault Injections with High Temperatures

Pedro Maat C. Massolino, Barış Ege, and Lejla Batina[1]

Digital Security Group - ICIS,
Radboud University,
Toernooiveld 212, 6525 EC Nijmegen, The Netherlands
emails: {p.massolino,b.ege,lejla}@cs.ru.nl

Abstract. Power and clock glitch attacks on smart cards can help an attacker to discover some internal secrets or bypass certain security checks. Also, an attacker can manipulate the temperature and supply voltage of the device, thus making the device glitch more easily. If these manipulations are within the device operating conditions, it becomes harder to distinguish between an extreme condition from an attacker. To demonstrate temperature and power supply effect on fault attacks, we perform several tests on an Atmega 163 microcontroller in different conditions. Our results show that this kind of attacks are still a serious threat to small devices, whilst maintaining the manufacturer recommendations.

1 Introduction

Clock or power glitch attacks try to interfere with a device clock or power interface, respectively, in order to jump/repeat instructions or insert wrong values. As shown by Schmidt and Herbst [11] it is possible to bypass the card security or obtain secret information by jumping instructions.

By experimenting with the precise timing of the glitch, the supply voltage and the ambient temperature, we can characterize the fault profile, or in other words, the fault parameters leading to a "successful" fault on the target. Where "successful" fault means faults that are able to change the device behavioral to the desired behavioral. This is of great importance as the device is expected to perform in different environmental conditions such as various ambient temperatures or supply voltages, and therefore it should be tested on resistance to fault injection in those conditions as well. In this way, the fault injection profile of the target can be fully determined and therefore the risk of missing a possible vulnerability is further reduced.

In this work, the main focus is to investigate when then it becomes easier (or harder) to induce faults on the device by powering from 3, 4 and 5 Volts and temperatures of 20°C, 40°C and 60°C. This is done in intermediate stages and results are presented in terms of the number of parameters which can be used to induce a successful glitch, either through clock or the power line.

The main contributions of this work can be summarized as follows:

- The minimum amplitude necessary for a glitch in the target clock line can change depending on if it is before or after the positive edge.
- When the supply voltage of the target is reduced, it becomes easier to introduce a clock glitch with lower glitch amplitudes.
- We confirm that, as reported before by Korak et al. [7], when the device is heated up the target becomes more vulnerable to clock glitches.
- When a power glitch is synchronized with the target system clock, the parameters for a successful glitch change depending on the relative timing of it with respect to the positive edge of the target clock.
- When the supply voltage of the target is reduced or increased, this has a direct and substantial effect on the parameters that can be used for inducing a successful power glitch.
- When the target is heated up, it becomes less sensitive to power glitches.

2 Related Works

Several fault attacks on smart cards and microcontrollers with or without smart card enclosing are reported in the literature. Here we cite and describe a few works, encompassing power and/or clock glitch attacks with or without temperature manipulation.

Agoyan et al. [1] give some background theory on how clock glitch attacks work. Korczyc and Krasniewski [8] performed a clock glitch attack on an AES implementation in an FPGA with small changes in the temperature. Their experiments show that with an increase of 3.5°C in surface temperature, the number of faulty bytes more than doubles.

[1] This work was supported in part by the Technology Foundation STW (project 13499 - TYPHOON & ASPASIA), from the Dutch government.

A more extensive fault injection analysis on an AES circuit was made by Zussa et al. [12]. In their analysis, both operational voltage and temperature were taken into account. They present the relations between the maximum operating frequency, the temperature and the supply voltage. They conclude that when the target device is heated up and underpowered, it becomes more sensitive to clock faults. However, power glitches in the same conditions were not performed.

Bareghi et al. [3] also perform clock glitch attacks on a different manner. They induce faults by underpowering the target device, rather than reducing the supply voltage for a short period of time. This approach showed that it was possible to introduce glitches on load instructions without affecting the clock line or generator.

Korak and Hoefler [6] also applied underpowering technique on their work. They combine underpowering for less than two clock cycles and clock glitches to introduce faults without being detected in an ARM Cortex–M0.

Another clock glitching experiments is by Korak et al. [7]. In this work the experiments are performed on an AVR ATmega162 microcontroller in a high temperature setting. The authors conclude that by heating up the target device, certain faults can be induced with more parameters than at room temperature, therefore making the target more vulnerable to clock glitches.

Although the aforementioned work shows a clear relation between temperature and fault injection attacks on the clock line, they do not consider other means of injecting faults, like power glitching. However, Carpi et al. [4] presents an analysis on algorithms to find suitable parameters for power glitches in an efficient way. Their results show their AdaptZoom algorithm requires the least number of iterations to find suitable glitch parameters.

When the approaches taken in the previous works are considered, this work is closer to the one presented by Carpi et al. [4]. However, this work focuses on finding the number of parameters leading to a successful fault, rather than finding a fault in the least number of experiments. Hence this work aims to provide a fault profile (for both clock and power faults) for a given target while taking into account different factors such as the ambient temperature and the supply voltage.

3 Target Microcontroller & Program

The target for our study case is a smart card with an AVR ATmega163 microcontroller. The main reason for this choice of platform is the ease of uploading and running code on the device, as well as the fact that it runs on an external clock. The external clock in particular makes the profiling stage much simpler, and therefore enables a thorough investigation of the glitch parameters.

The program running on the target microcontroller is a PIN verification program used by Carpi et al. [4]. It compares a PIN array stored in the internal memory with the input PIN. The comparison is done though a counter that is incremented for each correct digit. At the end, the number in the counter is verified and then the execution goes for the next instruction in case the PIN is correct. Otherwise, the execution jumps to the function that informs that the wrong PIN has been input.

For our fault injection attempts, we input a fixed, incorrect PIN to the smart card for each iteration. Therefore, when all PIN digits have been verified, since at least one is wrong, the device answer the input PIN is wrong. If a glitch is performed during this final verification, then it is possible to skip the verification and receive as answer that the input PIN is correct. This scenario is called a *successful glitch.* While, unsuccessful glitches can be a wrong PIN response that can cause the smart card to reset and return the reset message (ATR), or can lead to a communication failure due to an unexpected behavior in the execution process.

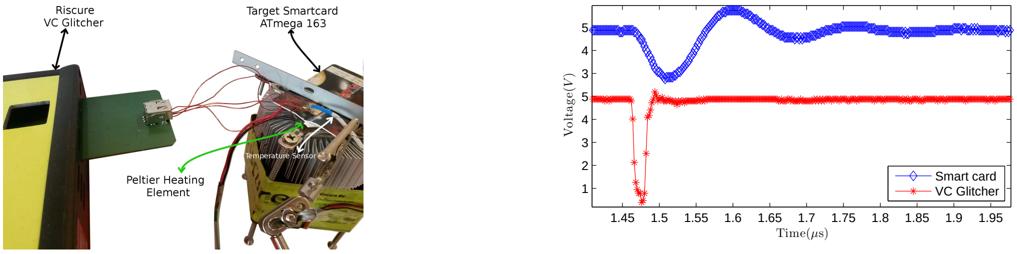
4 Fault Injection Setup

We use the Riscure VC Glitcher for our perturbation attempts as it provides reliable and reproducible faults on both power (VCC) and the clock (CLK) line of smart cards. Although the equipment can produce faults reliably, finding 'good' parameters for fault injection is still a non-trivial task. There are a few attempts in the literature to find 'good' glitch parameters besides using random search [4,10]. However, those works focus on finding a 'good' glitch parameter with the least amount of trials. Our goal is to profile the change in the number of 'good' parameters for fault injection rather than finding only one parameter which results in a successful glitch.

We use a Python script to search for the glitch length and the glitch voltage of as many 'good' glitches as we can find for a given timing for the glitch. The definitions of the parameters "glitch delay", "glitch length" and "glitch voltage" are as follows. "Glitch delay" is the time between the clock falling edge and the glitch occurrence. "Glitch length" is the amount of time the glitch keeps active. "Glitch voltage " is the amount of voltage that will be added or subtracted from the current voltage being applied.

The search algorithm focuses on different responses observed as a result of an initial parameters scan for fault injection. If there are different responses for a couple of parameters, we do a focused search in between those ranges of parameters to fully profile the so called 'sensitive region' of the target. In this aspect, the search algorithm we use is similar to the AdaptZoom algorithm defined in [4]. However, as we are interested in the number of glitch parameters leading to a successful glitch (i.e. the 'sensitive region'), we iterate our algorithm a number of times to systematically profile the sensitive region of the target, rather than stopping the algorithm after a certain number of measurements.

Figure 4 shows a sample output of our algorithm, where the green dots represent a normal response, a yellow dot represents a reset, a blue dot represents a communication error and a red dot represents a successful glitch. Note that the number of parameters that returns a successful glitch (a red dot) is referred to as the 'sensitive region' in the rest of the paper.



(a) Setup

(b) Glitch shape

Fig. 1: The fault injection setup with extension cables for maintaining communication with the target smart card (left). The original glitch shape (bottom) vs the delivered glitch shape (top) for power (VCC) glitching (right).

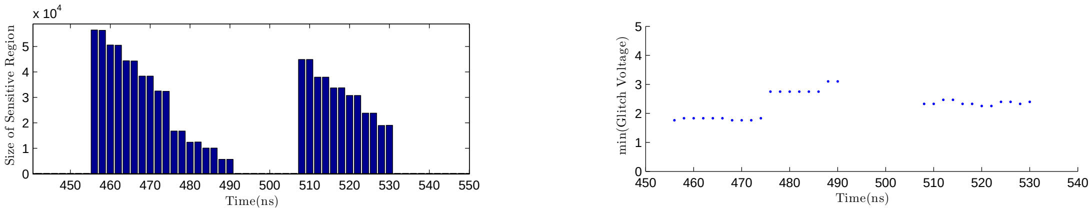
An important note regarding our fault injection setup is that we had to extend the connections of the VC Glitcher to be able to heat up the target without damaging the equipment that generates the glitches. A photo of our setup is given in Figure 1a. It should be noted that the extension using the additional wires, no matter how thin they are, introduces limitations on the shape of the glitch that we can deliver to the target. These limitations caused by the parasitical capacitance and resistance on the wire, which works as a low-pass filter to the glitch produced by the VC Glitcher. Figure 1b shows this difference when the glitch is measured at the VC Glitcher side (bottom plot), and when it is measured on the card (top plot). Hence, all experiments are done with those extension wires.

5 Perturbation Experiments

This section summarizes the observations we have made through various perturbation attempts with different parameters for timing, supply voltage and temperature. The results presented cover both clock and power glitching.

5.1 Effect of the Timing of the Glitch

One of the main goals of this research is to quantify the timing effect of a clock (CLK) and a power (VCC) glitch regarding the positive edge of the target clock cycle. Therefore, the results presented in this section are divided into two parts: results for CLK glitching and the results for VCC glitching. Both VCC and CLK glitching experiments presented in this section are performed with the target supplied with 5.0 V and 1 MHz.



(a) Sensitive region

(b) Minimum glitch amplitude

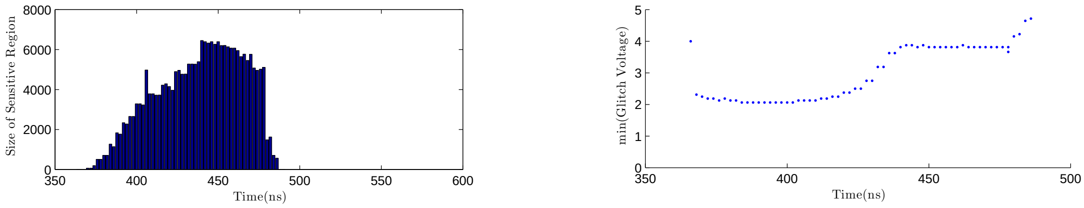
Fig. 2: The change in the size of the sensitive region of the target per given timing for clock (CLK) glitching (left). The minimum glitch amplitude required for a successful CLK glitch per given timing (right).

CLK glitching, as shown in Figure 2, can be done before or after the clock rising edge (500 ns), however glitching before shows to have more advantages than after.

Figure 2a shows the size of the glitch regions according to the glitch delay. Those regions after the rising edge are in less amount and lesser size when compared to regions before the rising edge. This illustrates that with CLK Glitching, it is easier to glitch before the rising edge than after.

Figure 2b shows the minimum glitch voltage absolute amplitudes values required to induce a successful glitch. The amplitudes are in absolute value, because before 500 ns, the glitch is a positive value, while after it becomes negative value. When comparing glitches before and after the clock rising edge, glitches before requires less voltage amplitude, therefore less power to be induced.

Given that glitches before the rising edge has greater chances to occur and requires less power, it is better to perform the attack before, than after. Similar to CLK glitching, VCC glitching is greatly affected by the precise glitch delay value. As shown on Figure 3a, the glitch region size changes alongside with the glitch delay. Also, it reveals that a successful VCC glitch is only possible before the positive edge of the target clock. This comparing with CLK glitching, VCC glitching still has more successful glitch delay values with our setup, as described in Section 4. However, if the glitch delay is fixed, CLK glitching has bigger glitch regions than VCC glitching.



(a) Sensitive region

(b) Minimum glitch amplitude

Fig. 3: The change in the size of the sensitive region of the target per given timing with VCC glitching (left). The minimum glitch amplitude required for a successful VCC glitch per given timing (right).

As much as the number of glitch parameters which can be used to induce a successful glitch are important, the minimum glitch voltage is also very important. If it is too low, it becomes harder for voltage sensors on smart cards to distinguish attacks from noise. Figure 3b presents the minimum glitch voltage required to induce at least one successful glitch with respect to various glitch delay values. Our experiments show that a lower glitch voltage was sufficient if the glitch is induced further away from the positive edge of the target clock.

Additionally, the location of the sensitive region also changes with respect to the exact glitch delay value. Figure 4 presents three sample plots that we get from our experiments for different glitch delay values. Our experiments show that a glitch induced earlier in the clock cycle requires a larger glitch length value. Therefore, as the glitch delay value increases, the glitch length necessary becomes lower. This translates as seen in Figure 4 in a moving effect of the the sensible region.

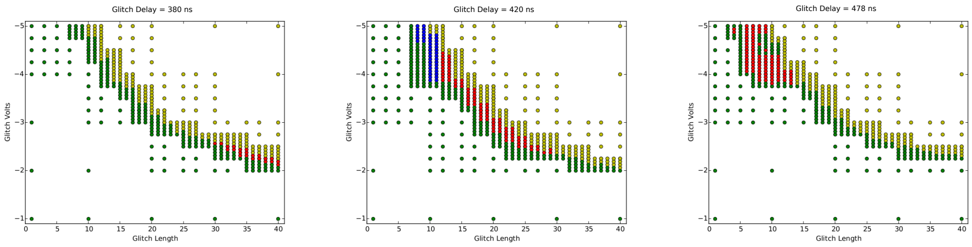


Fig. 4: The location of the sensitive region with respect to different glitch delay values: 380 ns (first), 420 ns (second), 478 ns (third).

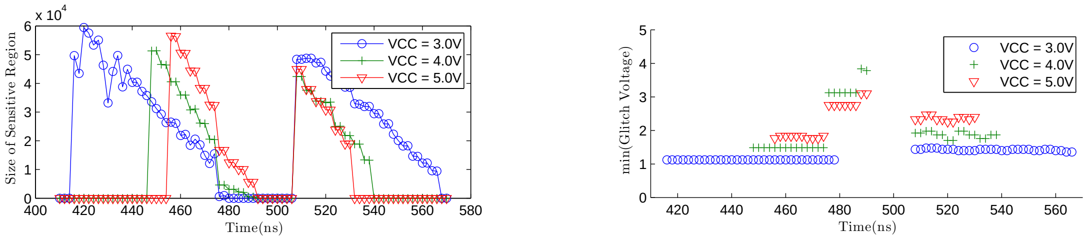
We do acknowledge that if the target device is not operating on an external clock, such analysis becomes more difficult. In such a case, first the internal system clock needs to be recovered, and only then such perturbation attempts can be made on a target which does not work on an external clock. However, O'Flynn and Chen propose ways to recover the internal clock of a target device [9]. Therefore we believe the information presented above can be of great importance for a complete security analysis of a target even if it is operating on an internal clock.

5.2 Effect of Supply Voltage

The effect of the supply voltage on clock glitches is also observed by Korak and Hoefler [6]. However, they investigate clock glitches while reducing the supply voltage for a short amount of time. Unlike their approach, we keep the supply voltage at a constant level which is well within the suggested operational parameters of the target microcontroller. Similar to the previous section, we present the experimental results in two separate parts: experiments on clock glitching and experiments on power glitching.

We analyze in our experiments with various supply voltage levels, the number of glitch parameters and the number of glitch delay values for successful clock glitches increases for lower supply voltage. The relation between clock glitches and the critical path of a circuit has already been explained in the literature by Agoyan et al. [1]. Figure 5a summarizes the results that we get from our setup. We see that lowering the supply voltage to 3 V makes a significant difference while experiments with 4 V supply voltage and 5 V supply voltage look very similar. These results are backed up by the fact that 4 V is the minimum supply voltage suggested by the manufacturer to run the microcontroller on 8 MHz. However, for supply voltage values between 2.7 V and 4 V, the suggested external clock frequency is 4 MHz [2].

Similar to the increase in the number of glitch parameters that lead to a successful glitch, also the minimum glitch value required for a successful glitch decreases if the voltage supplied to the microcontroller gets lower. We



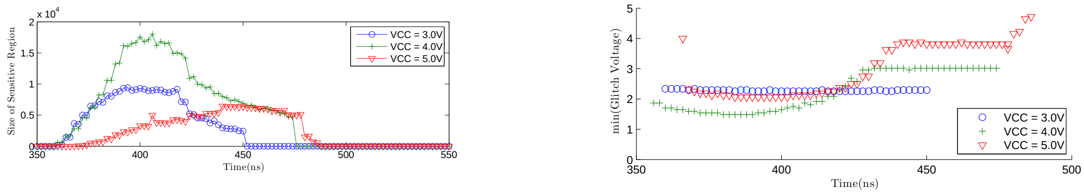
(a) Sensitive region

(b) Minimum glitch amplitude

Fig. 5: The change in the size of the sensitive region of the target per supply voltage (left). The minimum glitch amplitude required for a successful CLK glitch per supply voltage (right).

hypothesize that a clock glitch which uses a glitch delay value of less than 500 *ns* are interpreted as a positive clock edge in this case. Although the datasheet of ATmega163 suggests that only a minimum voltage of 2.1 V supplied for a duration of at least 100 *ns* can be interpreted as a positive clock, we see from Figure 5b that a value of 1.123 V is enough to inject a successful clock glitch when the supply voltage is fixed at 3 V.

Looking at clock glitching results, one might expect that inducing power glitches is easier with decreased supply voltage, since a lower supply voltage would also mean a longer critical path in the circuit. However, our experiments which are summarized in Figure 6a contradict this expectation. If the target microcontroller is powered with 3.0 V, the number of glitch delay values which lead to a successful glitch are smaller than the ones we get from experiments using higher supply voltages. Our experiments show that when the target is supplied with 4.0 V, the target becomes more vulnerable to power glitches. Moreover, Figure 6b shows that a supply voltage of 4.0 V also enables power glitching with the lowest glitch amplitudes. Also the characteristic trend we see in the minimum glitch voltage required for supply voltages 4.0 V and 5.0 V is not visible when the smart card is powered with only 3.0 V. Moreover, the effect that is shown in Figure 4 is not completely visible when the smart card is powered up with 3.0 V. Any glitch voltage value above −2.256 V does not result in a successful glitch. Since higher glitch values are possible when the smart card is powered with 4.0 V, we hypothesize that this behavior is connected to the fact that the microcontroller cannot run on an external clock faster than 4 MHz when the supply voltage is lower than 4.0 V.



(a) Sensitive region

(b) Minimum glitch amplitude

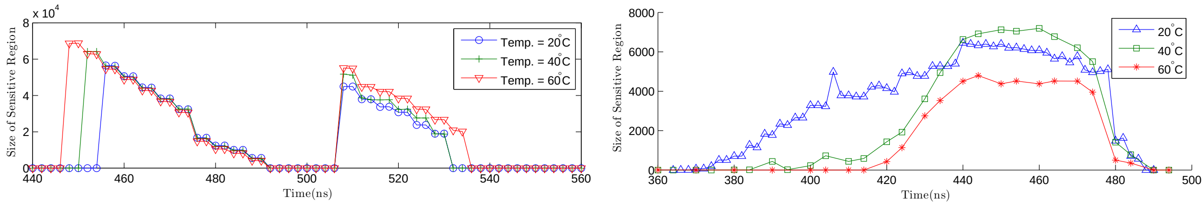
Fig. 6: The change in the size of the sensitive region of the target per supply voltage (left). The minimum glitch amplitude required for a successful VCC glitch per supply voltage (right).

5.3 Effect of the Temperature

This section investigates both clock glitching and power glitching when increasing the operating temperature. Even though there are previous works exploring the effect of increased ambient temperature on clock glitches [8,6], we present the results on clock glitching as well for the sake of completeness.

As explained in [5] (Chapter 12, p. 590), increasing the operating temperature of a circuit stretch its critical path. Orthogonally to the related works, Figure 7a shows the number of glitch parameters leading to a successful clock glitch when the operating temperature of the target is increased up to 60°C. Similar to the results presented by Korak et al. [6], the number of glitch delays that can lead to a successful glitch increases at high temperature. However, we do not heat up the device to 100°C [6] since the plastic coating of the target smart card starts deforming already at 80°C. We also observe from Figure 7a that the glitch delay values before and after the positive edge of the target clock (at 500*ns*) extend further to the lower temperatures delay values.

In contrast to clock glitching, in case of power glitches, increased temperature does not lead to a higher number of glitch delay values which can be used for inducing successful glitches. Figure 7b shows the number of glitch parameters leading to a successful glitch for power glitch when the target is supplied with 5.0 V. It can be deducted that 20 °C has the biggest amount of glitches, when considering possible delay values. However, for 40 °C it has the maximum amount of glitches in a exact delay value. Therefore, an attacker can exchange accuracy on the delay with the length and power of the glitch, with different temperatures. In case of even higher temperatures, as 60 °C, the amount of glitches per delay and the number of delays becomes lower than both 20 and 40 °C, thus not being a good option.



(a) CLK glitch

(b) VCC glitch

Fig. 7: CLK glitch area for each different delay at 5 Volts operation (left). VCC glitch area for each different delay at 5 V operation (right)

5.4 Reproducibility of the experiments

The experiments are repeated in two other cards of the same type in order to determine the reproducibility of the results presented in this paper. These experiments show that the actual number of parameters that can be used to inject faults vary between cards. However, it is possible to inject successful clock glitches with smaller glitch delay values with increased temperature. On the other hand, with power glitches, we observe that for each target, increased temperature leads to fewer glitch parameters to result in a successful glitch. Hence, the effects explained in the paper seem to be reproducible for the microcontroller used in the experiments.

6 Conclusions

We investigated how the number of glitch parameters leading to a successful fault changes with respect to the variations in the precise timing of the glitch, the supply voltage of the target, and the ambient temperature. We believe the analysis given is invaluable to a security analyst trying to assess the security of a device against fault attacks before it is deployed to the field.

Further, for the first time in the literature our results show that power glitches are negatively affected by increased ambient temperature. However, slightly lowering the supply voltage seems to increase the number of glitch parameters that can be used for inducing a successful glitch in the target. According to our analysis, even though heating up the target and lowering its supply voltage have similar effects on the critical path of the circuit, the effect of these methods on the number of successful glitch parameters do not seem to be the same. Hence, we believe that a detailed analysis of the relation between power glitches and the critical path of the target device would be invaluable to the field of fault analysis, as power glitches are a wide-spread threat to the security of embedded devices.

References

1. Agoyan, M., Dutertre, J.M., Naccache, D., Robisson, B., Tria, A.: When Clocks Fail: On Critical Paths and Clock Faults. In: Smart Card Research and Advanced Application, pp. 182–193. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2010), `http://dx.doi.org/10.1007/978-3-642-12510-2\_13`
2. Atmel Corporation: ATmega 163 Datasheet (2003), `http://www.atmel.com/Images/doc1142.pdf`
3. Barenghi, A., Bertoni, G., Parrinello, E., Pelosi, G.: Low Voltage Fault Attacks on the RSA Cryptosystem. In: Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on. pp. 23–31 (Sept 2009)
4. Carpi, R.B., Picek, S., Batina, L., Menarini, F., Jakobovic, D., Golub, M.: Glitch It If You Can: Parameter Search Strategies for Successful Fault Injection. In: Smart Card Research and Advanced Applications, Lecture Notes in Computer Science, vol. 8419, pp. 236–252. Springer International Publishing (2014), `http://dx.doi.org/10.1007/978-3-319-08302-5\_16`
5. Kaeslin, H.: Digital Integrated Circuit Design – From VLSI Architectures to CMOS Fabrication. Cambridge University Press (2008)
6. Korak, T., Hoefler, M.: On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms. In: Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on. pp. 8–17 (Sept 2014)
7. Korak, T., Hutter, M., Ege, B., Batina, L.: Clock Glitch Attacks in the Presence of Heating. In: Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on. pp. 104–114 (Sept 2014)
8. Korczyc, J., Krasniewski, A.: Evaluation of susceptibility of FPGA-based circuits to fault injection attacks based on clock glitching. In: Design and Diagnostics of Electronic Circuits Systems (DDECS), 2012 IEEE 15th International Symposium on. pp. 171–174 (April 2012)
9. OFlynn, C., Chen, Z.: Synchronous sampling and clock recovery of internal oscillators for side channel analysis and fault injection. Journal of Cryptographic Engineering 5(1), 53–69 (2015), `http://dx.doi.org/10.1007/s13389-014-0087-5`
10. Picek, S., Batina, L., Buzing, P., Jakobovic, D.: Fault Injection with a New Flavor: Memetic Algorithms Make a Difference. In: Constructive Side-Channel Analysis and Secure Design, Lecture Notes in Computer Science, vol. 9064, pp. 159–173. Springer International Publishing (2015), `http://dx.doi.org/10.1007/978-3-319-21476-4\_11`
11. Schmidt, J.M., Herbst, C.: A Practical Fault Attack on Square and Multiply. In: Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2008. pp. 53–58. IEEE (2008)
12. Zussa, L., Dutertre, J.M., Clédiere, J., Robisson, B., Tria, A.: Investigation of timing constraints violation as a fault injection means. In: 27th Conference on Design of Circuits and Integrated Systems (DCIS), Avignon, France (2012)