Fault Attack on FPGA implementations of Trivium Stream Cipher

F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández, M. Valencia-Barrero
Instituto de Microelectrónica de Sevilla/Universidad de Sevilla
Sevilla, España
Email: {potestad,cjesus}@imse-cnm.csic.es,{manolov}@dte.us.es

***Abstract*—This article presents the development of an experimental system to introduce faults in Trivium stream ciphers implemented on FPGA. The developed system has made possible to analyze the vulnerability of these implementations against fault attacks. The developed system consists of a mechanism that injects small pulses in the clock signal, and elements that analyze if a fault has been introduced, the number of faults introduced and its position in the inner state. The results obtained demonstrate the vulnerability of these implementations against fault attacks. As far as we know, this is the first time that experimental results of fault attack over Trivium are presented.**

***Keywords*—Fault Attack, Stream Cipher, Trivium, FPGA implementation.**

I. Introduction

The creation of secure cryptographic systems is in continuous progress. The emergence of new cryptographic algorithms entails the development of studies that attempt against the algorithm. Furthermore, the hardware implementation of the cryptographic algorithms has a second side of attack. To the classic attack over the algorithm, it must be added the attacks on the physical device that implements the cryptographic algorithm. Due to the information obtained during the operation of the circuit (power consumption, electromagnetic radiation, etc.) the device safety can be endangered. These attacks are named Side Channel Attacks [1], which can be of different types. On the one hand, there are Side Channel Analysis attacks, such as the Differential Power Analysis (DPA) and the Correlation Power Analysis (CPA), which attack the circuit through the power consumption measurements during circuit operation. On the other hand, the Active Fault Analysis attacks, such as Differential Fault Analysis (DFA) and Differential Fault Intensity Analysis (DFIA), which attack the circuit through the modification of the operation conditions to inject faults during the circuit operation.

Active Fault Analysis has been studied in literature since the work of Boneh et al [2] where they presented a fault attack on the RSA Cryptosystem. Since then, this technique has been applied successfully in many other cryptographic algorithms, including symmetric key systems like block ciphers and stream ciphers (Trivium among them).

In literature we can find several vulnerability analysis of the Trivium stream cipher against Active Fault Analysis attacks [3]-[8], but none of them checks its feasibility on a hardware implementation. Because of that, in this paper we present a behavioral analysis of FPGA implementations of Trivium cipher against fault injections in the clock signal and conclusions about the vulnerability of Trivium stream cipher against such attacks are extracted.

The rest of the paper is organized as follows. Section II introduces the architecture of the Trivium stream cipher and shows a review of the different theoretical Fault Injection techniques applied to the Trivium cipher. Section III presents the experimental Fault Injection Technique designed and in Section IV is presented the results obtained from the application of this technique against the Trivium stream cipher. Finally Section V presents the conclusions obtained from this work.

II. Fault Attacks on Trivium Stream Cipher

A. Trivium Stream Cipher

The Trivium stream cipher [9] is one of the finalists of the eSTREAM project. It is a synchronous cipher designed to generate up to 2^{64} bits of key stream from an 80 bits secret key and an 80 bits initialization vector (IV). The architecture of this cipher is based on three shift registers, 288 bits total, and combinational logic to provide its feedback. As others synchronous stream ciphers, the algorithm needs to be initialized with the load of the 288 bits of the shift registers (inner state) with one secret key, one initialization vector, zeros and ones. Before generating a valid key stream, the cipher needs to run during 1152 clock cycles. From that moment, it starts to generate a valid pseudo aleatory bits sequence (key stream).

The 288 bits of the inner state are distributed in three shift registers with different length. The first shift register has 93 bits, the second 83 bits and the third 111 bits. The feedback for each shift register is generated with AND and XOR operations. The key stream is the result of XOR operations on some bits in the shift register. Fig. 1 shows the schematic representation of the Trivium.

B. Fault Injection on Trivium

The Trivium stream cipher has been studied and analyzed theoretically to determine their vulnerability against fault injection side channel attacks. One of the first analysis is the work presented by Hojsík and Rudolf [3], where a Differential Fault Analysis (DFA) attack is done. This technique is a side channel attack in which an attacker is able to inject a fault in the encryption of decryption process. That is, the attacker is able to inject a fault over the cipher inner state. This technique assumes that an attacker is able to change only one bit of the inner state. In [3] two different techniques based on the

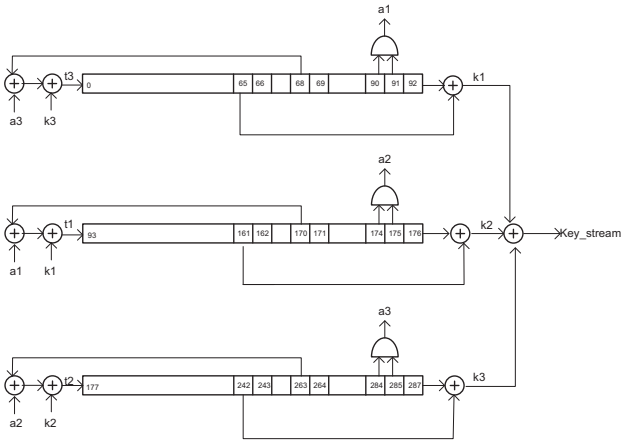

Fig. 1. Schematic representation of Trivium stream cipher.

same aim but using different mathematical formulations are presented. The second technique is the most efficient and they can retrieve the secret key with 43 fault injections. Later, the same authors present a new attack called Floating Fault Analysis of Trivium [4] where they reduce the number of fault injections to an average of 3.2. Yupu, Juntao, Qing y Yiwei [5] were based on those works to improve the cryptographic analysis and retrieve the secret key and initialization vector with an average of only 3.7 fault injections.

In [6], an Improved Differential Fault Analysis of Trivium is introduced, where they are able to retrieve the secret key with two fault injections over the inner state. A new analysis called Mutant Differential Fault Analysis of Trivium (MDFA), based on [3] is presented in [7]. They affirm that it is possible to break the system with only one fault injection. Other reference is the work called Improved Multi-Bit Differential Fault Analysis of Trivium [8], where an improvement over the system constraints presented in [5] is presented. Their method allows the attack on the system using different fault models, injecting a fault in an unknown cycle, and retrieving the secret key with four fault injections.

All these references coincide in the same assumption: in order to retrieve the secret key and initialization vector it is necessary to inject only one fault bit over any of the three registers of the stream cipher. This assumption establishes that if an attacker is able to do this injection, the cipher implementation will be vulnerable against fault injections attacks. However, none of these papers make experimental measurements of the fault injection possibilities. In the next section we present the experimental technique used to inject faults on the Trivium stream cipher, which involves injecting short pulses in the clock signal.

III. Fault Injection Technique Designed

As explained above, it is necessary the development of an experimental mechanism to inject faults in Trivium implementations and to measure the number of faults introduced. This mechanism will be implemented on FPGA devices and it is based on short pulses insertion in the clock signal. This technique is studied in [10] with theoretical analysis and practical experiments in AES block cipher, and in [11] where it is shown an attack made for fault injections with short pulses in the clock signal of different block ciphers. In [12] this technique is carried out to attack a FPGA implementation of the AES block cipher. Taking into account the theoretical models of vulnerability to fault injections in Trivium cipher reported in the literature, mentioned in Section II, the mechanism that we have developed must be able to insert only one fault in the inner state to be effective.

FPGAs have dedicated clock networks specially designed to distribute clock signals to all logic and IP blocks with low skew and latency. As a result, the synchronous circuits can work at very high frequencies if the combinational logic does not insert a big delay. From the first tests in Trivium implemented on FPGA, it was found that its maximum operating frequency coincides with the maximum operating frequency of the device. In the results shown in this paper, the device used is a Spartan 3E XC3S500E and the maximum operating frequency is 311 MHz. It has been also experimentally verified that the cipher Trivium does not work with a frequency of 317 MHz. Therefore, the insertion of a short pulse in the clock signal has the added difficulty that in order to inject faults in the cipher, the system must work slightly above their limit of operation.

Taking this into consideration, the system to be developed has three problems: *a)* frequencies well above the maximum can not be used, otherwise they will be filtered by the FPGA. *b)* Although, to insert a fault in the Trivium it is necessary a frequency slightly above the maximum, the whole system of control and measurement must operate at a frequency much lower than the maximum. *c)* We must ensure the faults are being introduced in the Trivium and we do not have any errors in the rest of the system.

Taking into account these problems, the developed system consists of a finite state machine that controls the operation of the Trivium (loading, operation and capture inner state). This state machine operates at a frequency much lower than the maximum operating frequency of the device. Independently, we generate a clock signal of a frequency able to make the Trivium fail. The state machine controls the generation of the pulse in the Trivium clock signal switching between a low frequency and the one that makes Trivium fail.

There are different ways to generate a short pulse in a signal. One of them is with a logical operation between two shifted signals. However, due to the particular structure of FPGA devices, it is very difficult to control the delays with the precision needed for this application. Thus, this method is not possible to be implemented in FPGA. Another method is to generate a high clock frequency and switch it with a low clock frequency. Implementing this technique in FPGA devices is possible thanks to their specific resources, to generate high frequencies clocks signals and to commute between two clock signals.

As mentioned before, the fault injection system is controlled by a state machine. This state machine control the Trivium stream cipher, namely, select and load the secret key and the initialization vector, generate n bits of the key stream and reset the cipher. On the other hand, this state machine also controls the subsystem for the short pulse generation, allowing

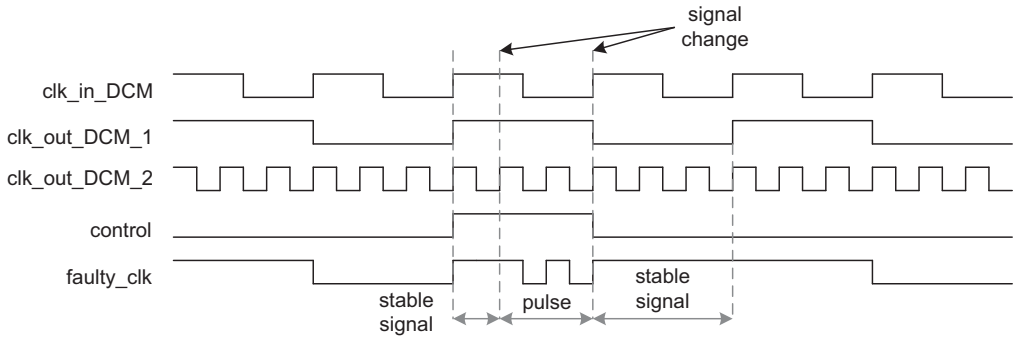

Fig. 2. Timing diagram to obtain the short pulse.

it to choose the clock cycle in which the pulse is injected. This last point is done to check the behavior of the cipher under different attack conditions. The clock frequency of the state machine is lower than the maximum operating frequency of the FPGA, so we are sure that the entire control system works without any error.

To generate the short pulse on the clock signal, Digital Clock Manager (DCM) available in Xilinx devices is used. The DCM can generate clocks with different frequencies from an input clock. The DCM does not only allow the generation of clocks with lower frequencies than the input clock, but also clocks with higher frequencies. In the developed system, the DCM has been used to generate a clock for the whole system and the Trivium with an optimal frequency, and to generate a clock whose frequency is above the maximum frequency of the device. Considering that the maximum frequency of the Spartan 3E XC3S500E device is 311 MHz, a clock of 316.66 MHz has been generated because was tested that the Trivium fails at that frequency. We have carried out several tests to verify that the clock signal of 316.66 MHz is being generated correctly. The input clock frequency is 50 MHz. With both clock signals, one of optimal frequency and the other with the fault frequency, the entire system can be controlled and the short pulse that allows the fault injection in the stream cipher can be generated. Switching between the two signals is done using a clock signal multiplexer, which allows switching between clock signals without generating additional pulses. Fig. 2 shows the timing diagram of the clock signal generation with the short pulse. To achieve that when switching between the two clocks introduce only one pulse of the fail clock, the frequency of the optimal clock must be a quarter of the fail clock.

IV. Results

The fault injections system has been designed in VHDL and implemented with ISE 14.7 Xilinx tool. The tests have been done over a FPGA Spartan 3E XC3S500E and the sampling of data has been done with ChipScope Pro Analyzer. To carry out the tests, the design of the Trivium cipher has been modified with the aim to have access to its inner state register. Moreover, one register has been introduced which is used to copy the Trivium inner state to analyze it later. To analyse the routing dependency and their behaviour against the same attack, three copies of the Trivium have been implemented in the device. The short pulse is injected in two of the Triviums, but the third Trivium always works properly. Its output is correct and it is used to compare it with the outputs of the others Triviums.

As already mentioned in this text, to break the system through fault attack, it is necessary to inject only one wrong bit on the inner state of the stream cipher. Taken this into account, it will be considered that a fault injection is effective or successful when a fail is injected in any of the three shift registers of the cipher and this injection is of only one wrong bit. In cases where the fault injections produce more than one wrong bit, this injection will be considered as not successful attack.

Considering that the timing simulations are not useful to know the fail frequency of the Trivium, a study was done about the different possibilities of the fault frequencies until the optimum frequency was found. Having been selected as optimal, the frequency at which it is possible to enter more effective faults. If the frequency increases slightly, the number of injected faults increases, so they are not effective.

We present results for four pairs of random key and IV, and the pulse has been injected in four different clock cycles. Firstly, Table I shows the number of faults injected for each pair key-IV in the four different clock cycles. When the number of faults is one, it means that for these conditions it has been injected only one wrong bit in the shift register and therefore the fault is effective. When the number of faults is zero, it means that we have not injected faults, and the cases where the number is bigger than one are the cases where the faults have been injected but these faults are non effective.

When the tests were repeated in the same situations (same key-IV and clock cycle), the number of injected faults can change slightly. Table II shows the different number of faults injected for the pair key 1 and IV 1 when the test is repeated a hundred times. Three different options are obtained. In two of them it is possible to inject faults (effective or non-effective) and, in the third option it is not possible to inject a fault. Each of these options has its own occurrence percentage, and the percentage of fault injection is always greater than non-fault injection. In addition, it can be seen that the Triviums do not fail at the same way in all cases, being possible to inject faults over each cipher through the change of the insertion cycle. The tests have been repeated for the other combinations of key and IV and the results are very similar.

TABLE I. FAULTS INJECTIONS ON THE CIPHERS FOR EACH KEY/IV PAIR AND INSERTION CYCLE.

	Key 1 IV 1		Key 1 IV 2		Key 2 IV 1		Key 2 IV 2	
Insertion cycles	Trivium 1	Trivium 2	Trivium 1	Trivium 2	Trivium 1	Trivium 2	Trivium 1	Trivium 2
1200	1	0	1	1	1	0	0	0
1300	2	1	1	1	1	0	0	0
1500	2-3	2	1	0	1	1	1	1
1750	1	1	1	0	1	0	1	0

TABLE II. NUMBER OF TYPE 1 AND TYPE 2 FAULTS INJECTED ON EACH CIPHER FOR DIFFERENT INSERTION CYCLES.

Key 1 IV 1	Fault Type 1			Fault Type 2			No fault
Insertion cycles	Trivium 1	Trivium 2	Occurrence	Trivium 1	Trivium 2	Occurrence	Occurrence
1200	1	0	72%	-	-	-	28%
1300	2	1	57%	2	0	9%	34%
1500	3	0	68%	2	2	5%	27%
1750	1	1	55%	1	0	6%	39%

TABLE III. POSITIONS OF THE INJECTED FAULTS TYPE 1 AND TYPE 2 ON EACH CIPHER.

Key 1 IV 1	Fault Type 1		Fault Type 2	
Insertion cycles	Trivium 1	Trivium 2	Trivium 1	Trivium 2
1200	2	-	-	-
1300	3/2	3	3/2	-
1500	96/3/2	-	96/3	3/2
1750	2	2	2	-

Table III shows the positions of the fault bits in the inner state for the same tests shown in Table II. Even though the bits that tend to fail oscillate around the same positions, they change their number and position for the same cipher, the same pair key-IV and different insertion cycle. Regarding the bits position, it is found that the bits that tend to fail are near the bits used by the Trivium for feedback or for logic operations.

Considering all tests performed, it is concluded that the developed system is able to introduce faults in the Trivium in 59% of the tests performed. Within this 59%, the efficiency of the designed system, or percentage of times where it has been injected an effective fault is 91.34%. This means that our system is able to introduce an effective fault in the 53.89% of the tests.

V. CONCLUSIONS

This work has presented one experimental system for fault injections on Trivium stream ciphers implemented on FPGA. With this system we have studied the Trivium stream cipher vulnerability against fault attack for pulse injections on the clock line. The results show that the system has an efficiency fault injection of 59%. Counting all the tests, the developed system is able to introduce an effective fault (single fault in the inner state) in nearly 54% of the tests.

From all results, we can insure that the designed system is able to inject only one fault bit in the inner state of the Trivium stream cipher, and these injections depend on the implementation of each stream cipher and the clock cycle in which they are inserted. Furthermore, it has been seen that the bits that tend to fail are those bits used to do the logical operations of the cipher. Another point of interest is the difference in the vulnerability in relation to the key and initialization vector pair and insertion cycle. The same key/IV pair in different clock cycles inject different faults and besides these bits change in relation to the key/IV.

ACKNOWLEDGMENT

This work was partially supported by the Spanish Ministry of Economy and Competitiveness (with support from the European Regional Development Fund - FEDER) under contracts CITIES (TEC2010-16870), CESAR (MEC TEC2013-45523-R), LACRE (CSIC 201550E039) and MISAL (CSIC).

REFERENCES

[1] S. Patranabis, *et al.*, "Using State Space Encoding To Counter Biased Fault Attacks on AES Countermeasures", In: COSADE 2015, 2015.

[2] D. Boneh, R.A. DeMillo, R.J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults", *Lecture Notes in Computer Science*, vol. 1233, pp. 3751, Springer, 1997.

[3] M. Hojsík and B. Rudolf, "Differential fault analysis of Trivium", *Fast Software Encryption*, pp. 158-172, Springer, 2008.

[4] M. Hojsík and B. Rudolf, "Floating fault analysis of Trivium", *Progress in Cryptology*, pp. 239-250, Springer, 2008.

[5] Y. Hu, *et al.*, "Fault analysis of Trivium", *Designs, Codes and Cryptography*, vol. 62, no 3, pp. 289-311, Springer, 2012.

[6] M.S.E. Mohamed and J. Buchmann, "Improved differential fault analysis of Trivium", In: COSADE 2011, pp. 147-158, 2011.

[7] M.S.E. Mohamed and J. Buchmann, "Mutant Differential Fault Analysis of Trivium MDFA", *Information Security and Cryptology-ICISC 2014*, pp. 433-446, Springer, 2014.

[8] P. Dey and A. Adhikari, "Improved Multi-Bit Differential Fault Analysis of Trivium", *Progress in Cryptology-INDOCRYPT*, pp. 37-52, Springer, 2014.

[9] C. De Canniere and B. Preneel, Trivium, A Stream Cipher Construction Inspired by Block Cipher Design Principles, eSTREAM, ECRYPT Stream Cipher Project.

[10] M. Agoyan *et al.*, "When clocks fail: On critical paths and clock faults", *Smart Card Research and Advanced Application*, pp. 182-193, Springer, 2010.

[11] T. Fukunaga, J. Takahashi, " Practical fault attack on a cryptographic LSI with ISO/IEC 18033-3 block ciphers", *Fault Diagnosis and Tolerance in Cryptography*, pp. 84-92, IEEE, 2009.

[12] Y. Ren, A. Wang, L. Wu, "Transient-Steady Effect Attack on Block Ciphers", *Cryptographic Hardware and Embedded Systems*, pp. 433-450, Springer, 2015.