Review of Breaking Trivium Stream Cipher Implemented in ASIC Using Experimental Attacks and DFA

F. E. Potestad-Ordóñez
IMSE-CSIC/U. Sevilla
potestad@imse-cnm.csic.es

E. Tena-Sánchez
IMSE-CSIC/U. Sevilla
erica@imse-cnm.csic.es

C. Fernández-García
IMSE-CNM-CSIC
carlos@imse-cnm.csic.es

V. Zuñiga-González
IMSE-CNM-CSIC
virginia@imse-cnm.csic.es

J. M. Mora-Gutiérrez
IMSE-CNM-CSIC
jmiguel@imse-cnm.csic.es

C. Baena-Oliva
IMSE-CSIC/U. Sevilla
cbaena@imse-cnm.csic.es

P. Parra-Fernández
IMSE-CSIC/U. Sevilla
pparra@imse-cnm.csic.es

A. J. Acosta-Jiménez
IMSE-CSIC/U. Sevilla
acojim@imse-cnm.csic.es

C. J. Jiménez-Fernández
IMSE-CSIC/U. Sevilla
cjesus@imse-cnm.csic.es

***Abstract*—In this paper, we present a review of the work [1]. In this work a complete setup to break ASIC implementations of standard Trivium stream cipher was presented. The setup allows to recover the secret keys combining the use of the active non-invasive technique attack of clock manipulation and Differential Fault Analysis (DFA) cryptanalysis. The attack system is able to inject transient faults into the Trivium in a clock cycle and sample the faulty output. Then, the internal state of the Trivium is recovered using the DFA cryptanalysis through the comparison between the correct and the faulty outputs. The secret key of the Trivium were recovered experimentally in 100% of the attempts, considering a real scenario and minimum assumptions.**

***Index Terms*—fault attack, Trivium, ASIC, DFA, key recovery.**

Tipo de contribución: Investigación ya publicada

I. Introduction

The Trivium stream cipher [2] was one of the eSTREAM project finalists and is part of the ISO/IEC 29192-3 [3] standard for lightweight stream ciphers. From an 80-bit secret key denoted as KEY and an 80-bit initialization vector denoted as IV, this cipher is able to generate in a synchronous way up to 2^{64} bits of key stream. Fig. 1 shows a schematic representation of Trivium internal structure. As it can be seen, its internal structure is performed by three shift registers comprising 288 bits in total and ten XOR gates and three AND gates for the feedbacks. Each of these three shift registers are composed by 93, 84 and 111 bits respectively. The KEY and the IV are loaded in the internal register, along with some prefixed zeros and ones. After the first 1152 clock cycles, the cipher generates a valid pseudorandom bit sequence. The key stream is the result of the XOR operations.

II. Cipher Vulnerability against DFA

A. Theoretical vulnerability

DFA is essentially a theoretical attack where if any attacker is able to inject transient faults into the operation of a device (either in its encryption or decryption processes) through the use of mathematical formulation, can obtain secret information contained in the device and thus endanger its security. Of the different assumptions necessary to carry out DFA on the Trivium stream cipher, the most important one is that the attacker is able to inject a single effective fault into the ciphers internal state and capture both the correct key stream and the one originated by that fault. For our system it is necessary to capture 800 bits. A more complete description of the mathematical aspects of the DFA system can be found in the references [4], [5]. In summary, and taking into account DFA nomenclature, the attacker is able to obtain both the key stream of the Trivium cipher $\{z_i\}_{i=1}^{\infty}$ produced by its internal state ISt_0 and the key stream produced from a transient random fault $\{z'_i\}_{i=1}^{\infty}$, introduced into the internal state which is now called $IS't_0$ because it contains the fault. The same attack and the capture of the faulty key stream must be carried out repeatedly under the same conditions: i.e., using the same key and IV and always attacking in the same clock cycle t_0.

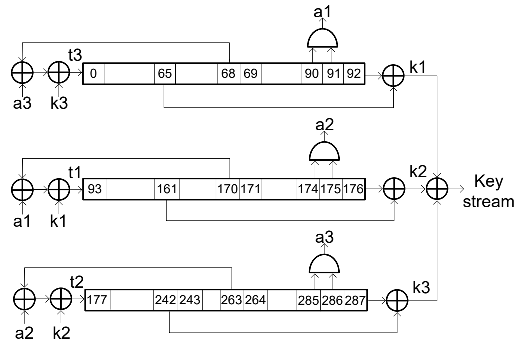

Fig. 1. Schematic representation of the Trivium stream cipher.

B. Experimental vulnerability

The results presented in [6] for attacks carried out by manipulating the clock signal in FPGA implementations of the Trivium showed that it is possible to inject faults only in flip-flops with feedback inputs: namely, position bits 0, 93 and 177 of the internal register or its neighbouring flip-flops. In addition, even with these flip-flops, faults can only be introduced in those that change their value. It is therefore only possible to obtain an average of three faulty key streams. Tests carried out on ASIC implementations showed the same behaviour: faults are injected into flip-flops whose inputs come from feedback or their neighbours. This is a serious problem because, in order to recover the internal state of the Trivium using the developed DFA, it is necessary to have more than three faulty key streams. Therefore, the developed

DFA system requires faulty key streams generated by faults injected in different positions for the same clock cycle. The fewer faulty key streams we have, the greater the brute force effort needed to recover the internal state of the cipher.

III. Experimental attack and results

The experimental attacks were carried out in a Trivium cipher implemented in a 90 nm ASIC technology. The key and IV to be used are loaded serially in the ASIC, and the clock and control signals of the Trivium are connected to the ASIC input pads. The key stream of the Trivium is connected to an output pad of the ASIC.

A. Attack Using Clock Glitches and how to achieve Multiple Faults in the Same Clock Cycle

To inject the faults into the cipher an active non-invasive fault injection system has been designed. It is based on inserting short pulses in the clock signal. This technique allows violating the setup times of the flip-flops making the sampled value on its output be erroneous which represents a fault injection in the Trivium cipher. The clock signal with the short pulses is externally generated and has to pass through the input pads of the integrated circuit to reach the circuit. This is a great challenge because low frequencies will not inject faults in the circuit, but very high frequencies can be filtered by the circuit pads.

It is possible to inject faults into more internal flip-flops because stream ciphers and Trivium in particular are built with shift registers. The shift registers make the fault injected into a flip-flop in one clock cycle appear as a fault injected in the next position of the shift register in the next clock cycle. If a fault is injected into the first bit of any of the three shift register (0, 93, 177), the faulty bit will not contribute to the key stream generation until it reaches one of the bits used for key stream generation (65, 161, and 242). During these clock cycles, the fault is only shifted through the shift register. Inject a fault in position 0, is therefore equivalent to introduce one fault in the position $0+n$, n clock cycles later. This increases the number of positions in which faults can be injected.

B. Results

To carry out the attack on the Trivium, we have used random keys and IVs. The attack cycle was set to $t_0 = 1332$, and the succession of attacks started in cycle $t_{20} = 1312$. Table I shows the results of the attack for Trivium. The table includes the number of the fault injection attempt, the fault injection cycle, the relative position of the injected fault (as if it was a fault inserted in cycle $t_0 = 1332$) and the number of bits of internal state retrieved by the DFA. The results show that with an average of 22 to 32 effective faults, it is possible to obtain the 288 bits of the internal state and therefore recover the secret key of the cipher. It should be noted that, in the case presented in Table I, after 25 attacks it would be feasible to break the cipher by brute force since most of the bits of the internal state are known.

TABLE I
Results obtained from the attack on Trivium.

F.I.A.[1]	C.C.[2]	R.P.[3]	B.R.[4]	F.I.A.	C.C.	R.P.	B.R.
1	1312	115	26	17	1296	131	–
2	1311	24	49	18	1295	132	229
3	1310	24	–	19	1294	134	234
4	1309	119	77	20	1293	134	–
5	1308	119	–	21	1292	135	250
6	1307	121	111	22	1291	220	257
7	1306	121	–	23	1290	222	274
8	1305	123	128	24	1289	222	–
9	1304	123	–	25	1288	139	279
10	1303	208	155	26	1287	147	281
11	1302	33	181	27	1286	249	282
12	1301	33	–	28	1285	143	285
13	1300	211	211	29	1284	143	–
14	1299	129	217	30	1283	228	285
15	1298	129	–	31	1282	145	287
16	1297	38	223	32	1281	230	288

[1] Fault Injection Attempt; [2] Clock Cycle of the attack; [3] Relative Position of the fault; [4] Number of bits Retrieved.

IV. Conclusions

This work describes the complete experimental breaking of Trivium ciphers implemented in ASIC technology. In 100% of the attempts, the secret key and IV were retrieved with minimal assumptions and in a real scenario. Firstly, experimental attacks were performed injecting a single fault into the internal register of the Trivium ciphers changing the external clock signal. Secondly, to inject faults in many positions of the internal register, we took advantage of the fact that the Trivium is built on the basis of shift registers. Thirdly, an inverse-operation Trivium was designed to get the secret key from a known internal state. The achievement of these three steps, together with the developed setup has allowed to obtain the secret key of the Trivium implemented in the ASIC. The work we have presented demonstrates that it is possible to experimentally break the security of ASIC implementations of the Trivium cipher using fault attacks and Differential Fault Analysis, in a short time and in a real scenario.

Acknowledgments

This work has been funded by project SCAROT 1380823-US/JUNTA/FEDER, UE. Thanks to SPIRS Project with Grant Agreement No. 952622 under the European Union's Horizon 2020 programme and Grant PID2020-116664RB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] F.E. Potestad-Ordóñez, M. Valencia-Barrero, C. Baena-Oliva, P. Parra-Fernández, C.J. Jiménez-Fernández, "Breaking Trivium Stream Cipher Implemented in ASIC Using Experimental Attacks and DFA". in *Sensors*, vol. 20, num. 6909, pp. 1-19, 2020.

[2] C.D. Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles". in *Proceedings of the 9th International Conference on Information Security (ISC'06)*, pp. 171–186, 2006.

[3] International Organization for Standardization: ISO/IEC 29192-3:2018. in *Information Security—Lightweight Cryptography—Part 3: Stream Ciphers*, International Organization for Standardization: Geneva, Switzerland, 2018.

[4] M. Hojsík, B. Rudolf, "Differential Fault Analysis of Trivium." in *Proceedings of the International Workshop on Fast Software Encryption (FSE'08)*, pp. 158–172, 2008.

[5] Y. Hu, J. Gao, Q. Liu, Y. Zhang, "Fault analysis of Trivium." in *Des. Codes Cryptogr*, vol. 62, pp. 289–311, 2012.

[6] F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "Vulnerability Analysis of Trivium FPGA Implementations." in *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, pp. 3380–3389, 2017.