Inapplicability of Fault Attacks against Trivium on a Cellular Automata Based Stream Cipher

Jimmy Jose[1] Sourav Das [2] Dipanwita Roy Chowdhury[1]

[1]Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur, India

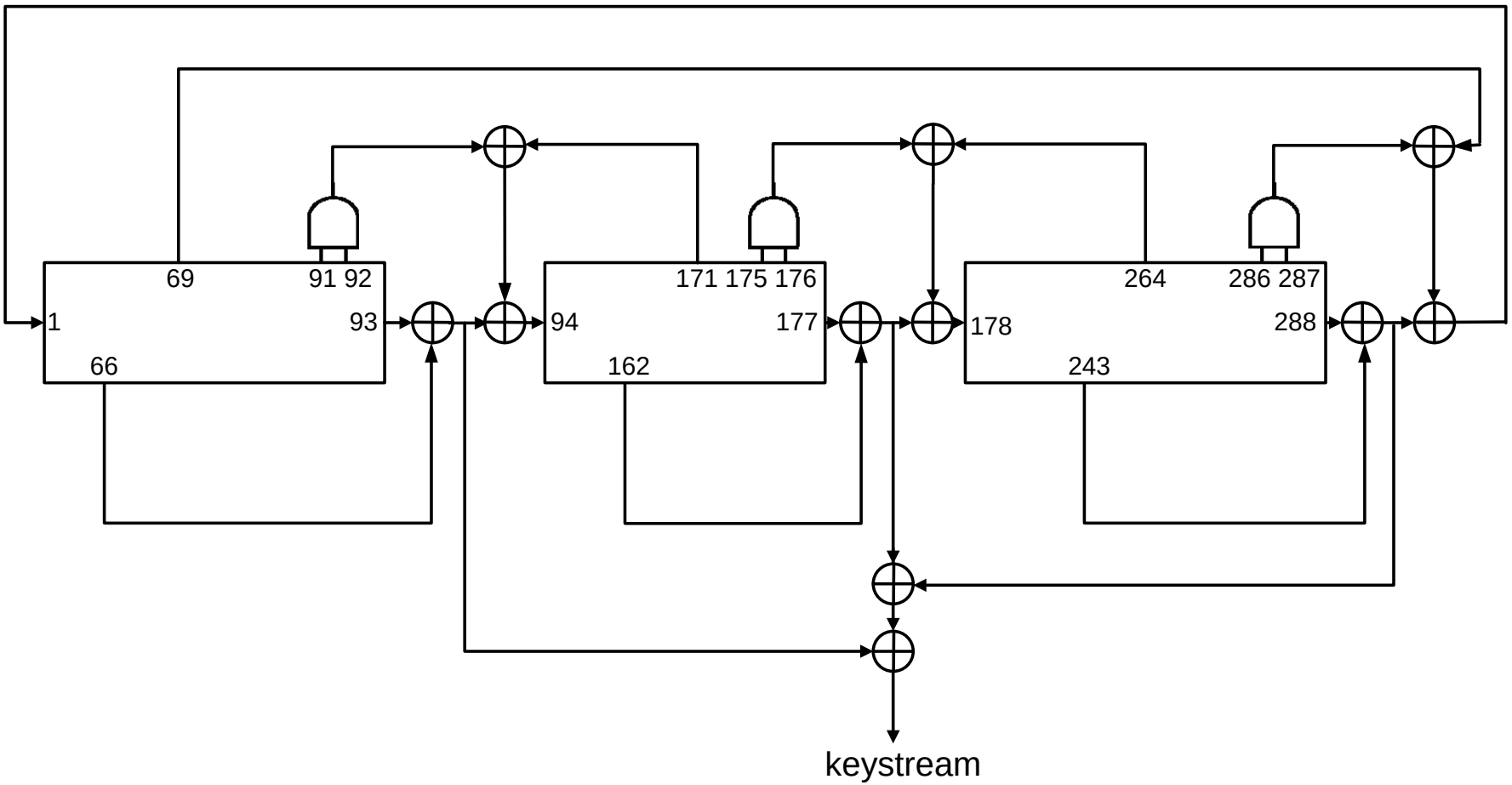
[2]Infinera India Pvt Ltd

September 25, 2014

Introduction

- Fault Attacks against Trivium [1] exploit
  - slow pace of non-linearisation
  - reversibility of encryption function
- CASTREAM [2], CA based Trivium-like cipher
  - fast non-linearisation
  - difficult to reverse
- We show
  - CASTREAM strong against fault attacks for which Trivium is vulnerable

Trivium[1]


69
91 92
1
93
66
171 175 176
94
177
162
264
286 287
178
288
243
keystream

Algorithm 1 Trivium Key Recovery

input: Trivium inner state $(s_1, \cdots, s_{288})$ at some time instant and IV $(v_1, \cdots, v_{288})$

output: Key $(k_1, \cdots, k_{80})$

1: **while** $(s_{81}, \cdots, s_{93}, s_{174}, \cdots, s_{288}) \neq (0, \cdots, 0, 1, 1, 1)$ **and** $(s_{94}, \cdots, s_{173}) \neq (v_1, \cdots, v_{80})$ **do**

2: $(t_3, s_1, \cdots, s_{92}) \leftarrow (s_1, s_2, \cdots, s_{93})$

3: $(t_1, s_{94}, \cdots, s_{176}) \leftarrow (s_{94}, s_{95}, \cdots, s_{177})$

4: $(t_2, s_{178}, \cdots, s_{287}) \leftarrow (s_{178}, s_{179}, \cdots, s_{288})$

5: $s_{93} \leftarrow t_1 \oplus s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{171}$

6: $s_{177} \leftarrow t_2 \oplus s_{162} \oplus s_{175} \cdot s_{176} \oplus s_{264}$

7: $s_{288} \leftarrow t_3 \oplus s_{243} \oplus s_{286} \cdot s_{287} \oplus s_{69}$

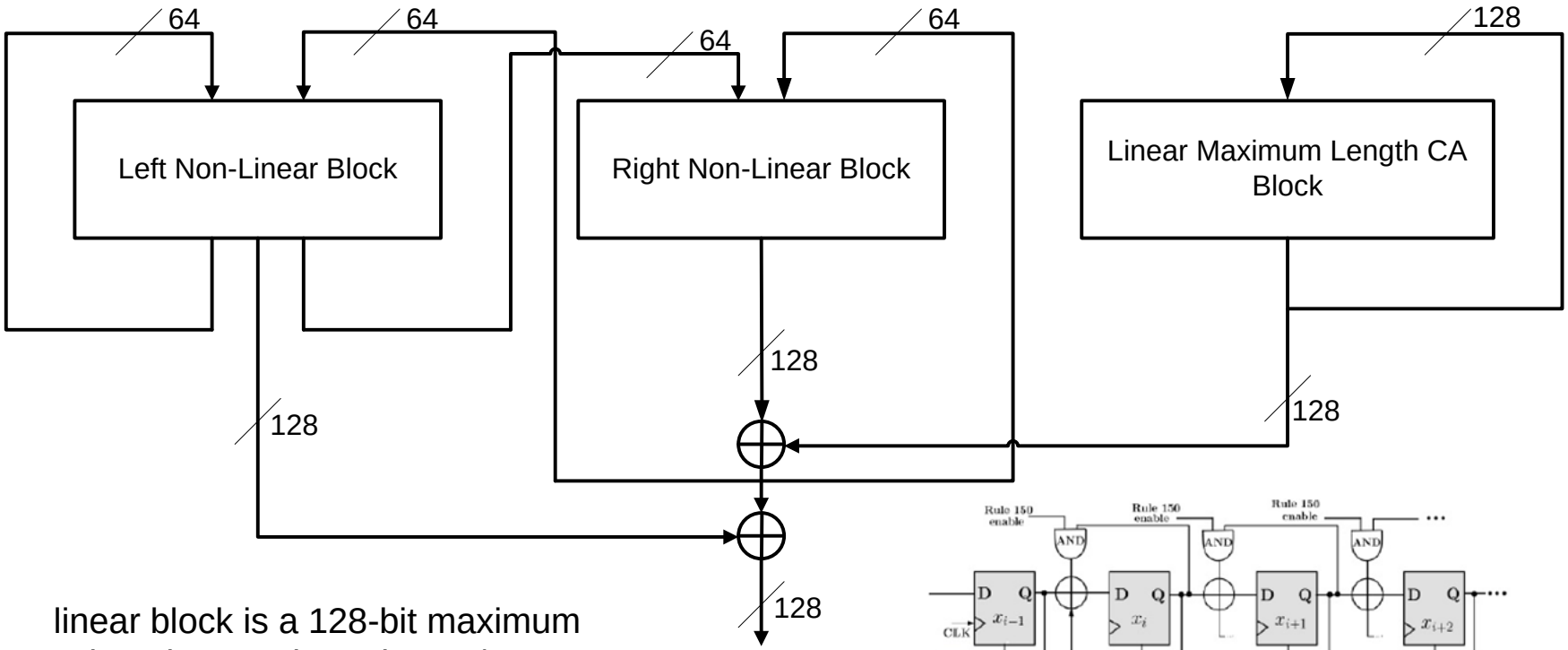
8: **end while**

9: $(k_1, \cdots, k_{80}) \leftarrow (s_1, \cdots, s_{80})$

Trivium Characteristics

- 66 linear, 82 quadratic equations
- the distance between state bits which contribute to keystream generation is different in each register
  - provides mechanism to find out fault injection position

CASTREAM[2]



linear block is a 128-bit maximum length CA using rules 90/150

4 cell linear hybrid CA based on rules 90, 150

Fault Attacks against Trivium and prevention in CASTREAM

1. Differential Fault Analysis (DFA)[3]

Attack 1

- linear equations and sufficient number of keystream difference equations represented as matrix
- solve matrix to get inner state

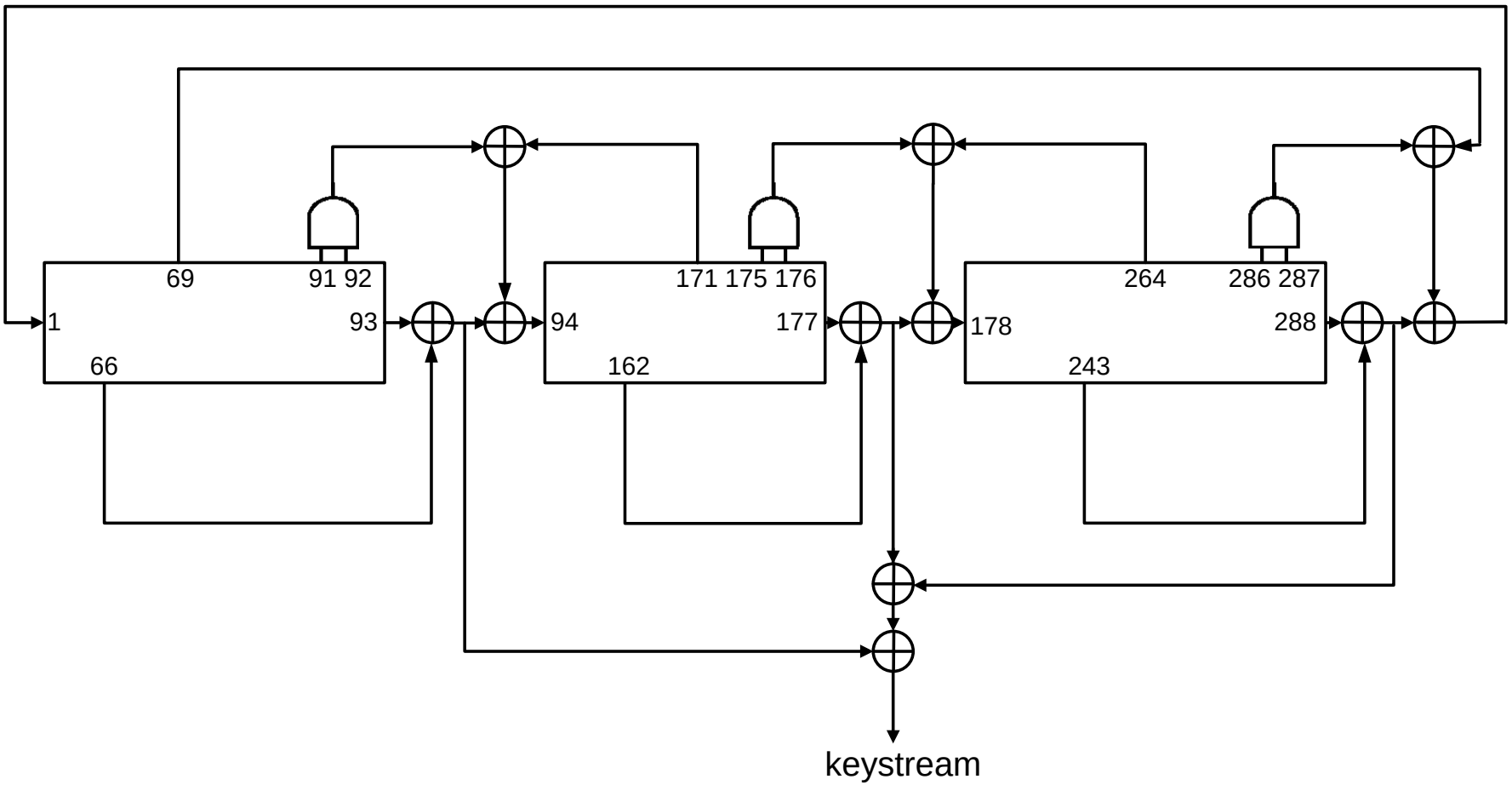
Attack 2

- pair-quadratic equations added with the equations in attack 1

Prevention in CASTREAM

- linear and pair-quadratic equations will not be generated
- multiple variable length s-boxes implemented with CA make function difficult to reverse
- single keystream bit in $(j+1)^{th}$ iteration depends on 77 bits of iteration j

Trivium[1]


69
91 92
1
93
66
171 175 176
94
177
162
264
286 287
178
288
243
keystream

2. Floating Fault Analysis (FFA)[4]

Difference from DFA

- treats each new inner state bit as a new variable
- 3.2 injections on average to break the cipher
- more number of variables revealed in each fault injection

Prevention in CASTREAM

- smallest s-box in a layer of multiple variable length s-boxes has a degree 3
- CASTREAM non-linear block has two layers of this kind
- linear or quadratic equations will not be generated

3. Hu et al. Fault Attack[5]

Attack Description

- Weaker assumptions in comparison with FFA
- Checking Method used to find out Fault Injection Time and Fault Position
- needs only original keystream and 16 fault injected keystreams of 195 bits length

Prevention in CASTREAM

- Checking Method not applicable as all inner state bits contribute to keystream bit
- linear and pair-quadratic equations cannot be generated
- algebraic degree of CASTREAM is 9

4. Mohamed et al. Attack[6]

Attack Description

- attack improves FFA by
  - improving equation preprocessing part
  - using SAT solver to speed up the solving part

Prevention in CASTREAM

- SAT solver uses linear and quadratic equations
- CASTREAM will not produce these as algebraic degree is 9
- CASTREAM is difficult to reverse

Conclusion

- studied fault attacks for which Trivium is vulnerable
- analysed the strength of CA based stream cipher CASTREAM against these attacks
- CASTREAM has been shown to prevent these attacks by exploiting the inherent properties of CA

References


De Canniere, C., Preneel, B.: Trivium Specifcation, accessed 13 September 2014, http://www.ecrypt.eu.org/stream/triviump3.html


Das, S., RoyChoudhury, D.: CASTREAM: A New Stream Cipher Suitable for Both Hardware and Software. In: ACRI 2012. LNCS, vol. 7495, pp. 601–610, Springer, Heidelberg(2012)


Hojsik, M., Rudolf, B.: Differential FaultAnalysis of Trivium. In: FSE 2008. LNCS, vol. 5086, pp. 158-172, Springer, Heidelberg(2008)


Hojsik, M., Rudolf, B.: Floating Fault Analysis of Trivium. In: INDOCRYPT 2008. LNCS, vol. 5365, pp. 239-250, Springer, Heidelberg(2008)


Hu, Y., Gao, J., Liu, Q., Zhang, Y.: Fault analysis of Trivium. Design Code and Cryptograpy. 62:289-311 DOI 10.1007/s10623–011–9518–9(2011)


Mohamed,M. S. E., Bulygin, S., Buchmann, J.: Improved Differential Fault Analysis of Trivium. In: Proceedings of COSADE 2011, pp. 147 – 158(2011)


Das, S., RoyChoudhury, D.: Generating Cryptographically Suitable Non-linear Maximum Length Cellular Automata. In: ACRI 2010. LNCS, vol. 6350, pp. 241–250, Springer, Heidelberg(2010)

Thank You