FResCA: A Fault-Resistant Cellular Automata Based Stream Cipher

Jimmy Jose[1,2] Dipanwita Roy Chowdhury[1]

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

[2]Department of Computer Science and Engineering,
National Institute of Technology Calicut, India

September 5, 2016

eSTREAM project

- eSTREAM project [1] - an effort to find out compact efficient stream ciphers
- divided into two categories:
  - software oriented
    - fast encryption in software
  - hardware oriented
    - fast encryption with less hardware

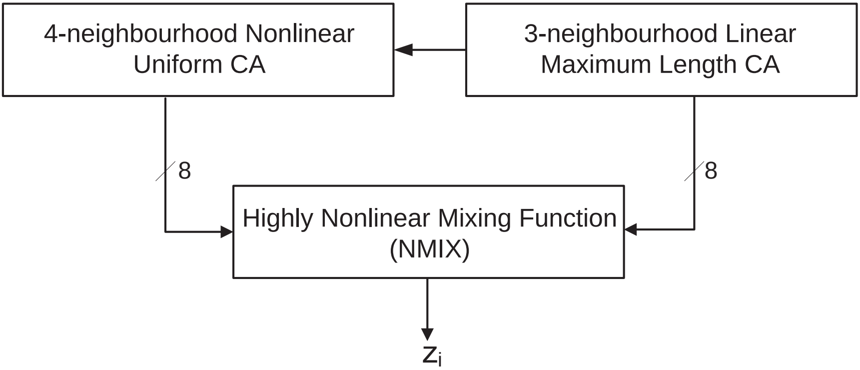
ACRI 2014

- Trivium cipher is a hardware oriented eSTREAM finalist
- Inapplicability of fault attacks against Trivium on a cellular automata based stream cipher was shown in ACRI 2014 [2]

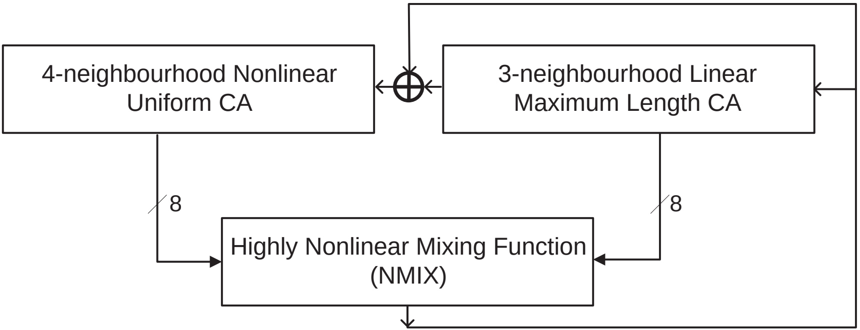
Fault Resistant Cipher

- Grain, another hardware oriented eSTREAM finalist, is vulnerable to fault attacks [3, 4]
- We have shown that 4-neighbourhood CA has good cryptographic properties [5]
- We design a fault-resistant 4-neighbourhood CA based Grain-like cipher

Fault-Resistant CA Based Stream Cipher - FResCA



FResCA intialisation

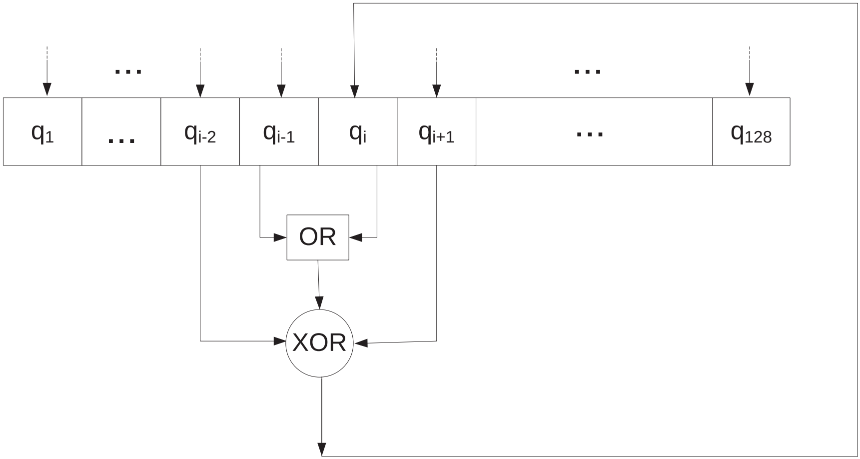


FResCA - Nonlinear Block

1	2	...	128
43350	43350	...	43350

- Rule 43350: $q_i(t+1) = q_{i-2}(t) \oplus q_{i+1}(t) \oplus (q_{i-1}(t) + q_i(t))$

FResCA - Rule 43350



- Rule 43350: $q_i(t+1) = q_{i-2}(t) \oplus q_{i+1}(t) \oplus (q_{i-1}(t) + q_i(t))$

FResCA - Linear Block

1	2	⋯	28	29	30	⋯	128
150	90	…	90	150	90	…	90

- Rule 90: $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$
- Rule 150: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$

nonlinear mixing is achieved by using NMIX function [6]
two n-bit inputs $X = (x_{n-1}x_{n-2}\cdots x_0)$, $Y = (y_{n-1}y_{n-2}\cdots y_0)$, and output $Z = (z_{n-1}z_{n-2}\cdots z_0)$
$z_i \leftarrow x_i \oplus y_i \oplus c_{i-1}$
$c_i \leftarrow x_0y_0 \oplus \cdots \oplus x_iy_i \oplus x_{i-1}x_i \oplus y_{i-1}y_i$
and $x_{-1} = y_{-1} = c_{-1} = 0, 0 \leq i \leq n-1$
c_i - carry term propagated from i-th bit position to $(i+1)$-st bit position

Working - FResCA cipher

- initialisation phase 32 cycles
- taps positions 1, 22, 43, 64, 65, 86, 107, and 128
- 32 cycles sufficient for all the 256 state bits to influence the cipher output

Nonlinear Mixing Block

nonlinear bits - $(b_1, \cdots, b_{128})$
linear bits - $(s_1, \cdots, s_{128})$
$z_1 = b_{128} \oplus s_{128} \oplus b_1 s_1 \oplus b_{22} s_{22} \oplus b_{43} s_{43} \oplus b_{64} s_{64} \oplus b_{65} s_{65} \oplus b_{86} s_{86} \oplus b_{107} s_{107} \oplus b_{86} b_{107} \oplus s_{86} s_{107}$
8 variables from the nonlinear block and 8 variables from the linear block
nonlinearity - 32256
correlation immunity - 1
resiliency - 1
algebraic degree - 2

Nonlinear Mixing Block

$$z_2 = b_{126} \oplus b_{127} \oplus b_{128} \oplus s_{127} \oplus (b_1 s_1) \oplus (b_1 s_2) \oplus (b_{105} b_{84}) \oplus (b_{105} b_{85}) \oplus (b_{105} b_{86}) \oplus (b_{105} b_{87}) \oplus (b_{105} s_{106}) \oplus (b_{105} s_{108}) \oplus (b_{106} b_{84}) \oplus (b_{106} b_{85}) \oplus (b_{106} b_{86}) \oplus (b_{106} b_{87}) \oplus (b_{106} s_{106}) \oplus (b_{106} s_{108}) \oplus (b_{107} b_{84}) \oplus (b_{107} b_{85}) \oplus (b_{107} b_{86}) \oplus (b_{107} b_{87}) \oplus (b_{107} s_{106}) \oplus (b_{107} s_{108}) \oplus (b_{108} b_{84}) \oplus (b_{108} b_{85}) \oplus (b_{108} b_{86}) \oplus (b_{108} b_{87}) \oplus (b_{108} s_{106}) \oplus (b_{108} s_{108}) \oplus (b_{127} b_{128}) \oplus (b_2 s_1) \oplus (b_2 s_2) \oplus (b_{20} s_{21}) \oplus (b_{20} s_{23}) \oplus (b_{21} s_{21}) \oplus (b_{21} s_{23}) \oplus (b_{22} s_{21}) \oplus (b_{22} s_{23}) \oplus (b_{23} s_{21}) \oplus (b_{23} s_{23}) \oplus (b_{41} s_{42}) \oplus (b_{41} s_{44}) \oplus (b_{42} s_{42}) \oplus (b_{42} s_{44}) \oplus (b_{43} s_{42}) \oplus (b_{43} s_{44}) \oplus (b_{44} s_{42}) \oplus (b_{44} s_{44}) \oplus (b_{62} s_{63}) \oplus (b_{62} s_{65}) \oplus (b_{63} s_{63}) \oplus (b_{63} s_{64}) \oplus (b_{63} s_{65}) \oplus (b_{63} s_{66}) \oplus (b_{64} s_{63}) \oplus (b_{64} s_{64}) \oplus (b_{64} s_{65}) \oplus (b_{64} s_{66}) \oplus (b_{65} s_{63}) \oplus (b_{65} s_{64}) \oplus (b_{65} s_{65}) \oplus (b_{65} s_{66}) \oplus (b_{66} s_{64}) \oplus (b_{66} s_{66}) \oplus (b_{84} s_{85}) \oplus (b_{84} s_{87}) \oplus (b_{85} s_{85}) \oplus (b_{85} s_{87}) \oplus (b_{86} s_{85}) \oplus (b_{86} s_{87}) \oplus (b_{87} s_{85}) \oplus (b_{87} s_{87}) \oplus (s_{106} s_{85}) \oplus (s_{106} s_{87}) \oplus (s_{108} s_{85}) \oplus (s_{108} s_{87}) \oplus (b_{105} b_{85} b_{86}) \oplus (b_{106} b_{107} b_{84}) \oplus (b_{106} b_{107} b_{85}) \oplus (b_{106} b_{107} b_{86}) \oplus (b_{106} b_{107} b_{87}) \oplus (b_{106} b_{107} s_{106}) \oplus (b_{106} b_{107} s_{108}) \oplus (b_{106} b_{85} b_{86}) \oplus (b_{107} b_{85} b_{86}) \oplus (b_{108} b_{85} b_{86}) \oplus (b_{21} b_{22} s_{21}) \oplus (b_{21} b_{22} s_{23}) \oplus (b_{42} b_{43} s_{42}) \oplus (b_{42} b_{43} s_{44}) \oplus (b_{63} b_{64} s_{63}) \oplus (b_{63} b_{64} s_{65}) \oplus (b_{64} b_{65} s_{64}) \oplus (b_{64} b_{65} s_{66}) \oplus (b_{85} b_{86} s_{85}) \oplus (b_{85} b_{86} s_{87}) \oplus (b_{106} b_{107} b_{85} b_{86})$$

Nonlinear Mixing Block

z_2 contains 26 variables from the nonlinear block and 15 variables from the linear block
algebraic degree increases from 2 to 4

Security of FResCA

Meier-Staffelbach Attack

Exploits the many-to-one mapping from the right-hand initial states to the temporal sequence or its right adjacent sequence

We have shown a class of 4-neighbourhood CA resists the attack [5]. The nonlinear rule of the cipher is from that class.

Security of FResCA

Linear Attacks

Derives linear approximations from the nonlinear relationships in the cipher

First output bit in initialisation phase has nonlinearity 32256. Nonlinearity increases with each iteration. Keystreams available from 33rd iteration only.

Security of FResCA

Correlation Attacks

Exploit the statistical weakness of the Boolean function

Nonlinear CA rule 43350 exhibits good correlation immunity. NMIX function guarantees correlation immunity and balancedness.

Security of FResCA

Algebraic Attack

Forms a system of multivariate equations. They are eventually solved to break the system

Ciphers having higher algebraic degree in their Boolean function are resistant to these attacks. Rate of increase in algebraic degree is high with each iteration in the cipher.

Security of FResCA

Scan-Based Side Channel Attacks

The reversibility of the algorithm is used to retrieve the key.

The combination of nonlinear and linear CA rules makes the cipher non-reversible.

NIST test

NIST test suite [7] is used for testing the randomness of the sequences generated by pseudo-random sequence generators.

Here, two key-IV pairs used. They are
Key: 0xFEDCBA98765432100123456789ABCDEF
IV : 0x0123456789ABCDEFFEDCBA9876543210
and
Key: 0x0123456789ABCDEFFEDCBA9876543210
IV : 0xFEDCBA98765432100123456789ABCDEF

NIST test

0.1 billion keystream bits are generated. The keystream is given as input to the test suite. The test suite was allowed to partition the input into 100 bitstreams where each bitstream contains 1 million bits.

NIST test

Table: NIST Test Results - FResCA with first key-IV pair

Test	P-val [1]	Proportion [2]	Test	P-val	Proportion
Frequency	0.4944	98/100	OverlappingTemplate	0.7792	100/100
Block Frequency	0.5341	100/100	Universal	0.3669	98/100
Cumulative Sums*	0.2897	98/100	Approximate Entropy	0.3345	99/100
Runs	0.7197	100/100	RandomExcursions*	0.7792	48/50
Longest Run	0.2493	100/100	RandomExcursionsVariant*	0.9835	49/50
Rank	0.2248	100/100	Serial*	0.9915	100/100
FFT	0.6371	100/100	LinearComplexity	0.2023	99/100
NonOverlappingTemp.*	0.7792	97/100			
*Note: In case of tests with more than one subset, the one with lowest proportion is shown here. F corresponds to failure.					

[1] probability that a perfect RNG should have generated a sequence which is less random than the sequence under test

[2] proportion of the sequence that pass the test

NIST test

Table: NIST Test Results - FResCA with second key-IV pair

Test	P-val	Proportion	Test	P-val	Proportion
Frequency	0.7598	99/100	OverlappingTemplate	0.2622	99/100
Block Frequency	0.6163	99/100	Universal	0.6282	98/100
Cumulative Sums*	0.6993	99/100	Approximate Entropy	0.4944	99/100
Runs	0.8343	100/100	RandomExcursions*	0.3345	56/57
Longest Run	0.3669	100/100	RandomExcursionsVariant*	0.2248	56/57
Rank	0.0146	99/100	Serial*	0.8343	99/100
FFT	0.7598	100/100	LinearComplexity	0.0590	100/100
NonOverlappingTemp.*	0.0966	96/100			

*Note: In case of tests with more than one subset, the one with lowest proportion is shown here.
F corresponds to failure.

NIST test

Table: NIST Test Results - variant of FResCA with first key-IV pair

Test	P-val	Proportion	Test	P-val	Proportion
Frequency	0.9241	100/100	OverlappingTemplate	0.4944	100/100
Block Frequency	0.9241	99/100	Universal	0.4750	100/100
Cumulative Sums*	0.9998	100/100	Approximate Entropy	0.0628	100/100
Runs	0.6787	100/100	RandomExcursions*	0.0742	58/60
Longest Run	0.0554	99/100	RandomExcursionsVariant*	0.1110	59/60
Rank	0.2133	99/100	Serial*	0.1373	98/100
FFT	0.4012	98/100	LinearComplexity	0.5749	99/100
NonOverlappingTemp.*	0.6371	96/100			
*Note: In case of tests with more than one subset, the one with lowest proportion is shown here. F corresponds to failure.					

NIST test

Table: NIST Test Results - variant of FResCA with second key-IV pair

Test	P-val	Proportion	Test	P-val	Proportion
Frequency	0.1917	100/100	OverlappingTemplate	0.8343	97/100
Block Frequency	0.7598	100/100	Universal	0.1154	100/100
Cumulative Sums*	0.4373	99/100	Approximate Entropy	0.6163	100/100
Runs	0.7399	99/100	RandomExcursions*	0.9850	61/62
Longest Run	0.7197	100/100	RandomExcursionsVariant*	0.1481	61/62
Rank	0.4750	100/100	Serial*	0.0329	98/100
FFT	0.5955	99/100	LinearComplexity	0.4012	100/100
NonOverlappingTemp.*	F	F			

*Note: In case of tests with more than one subset, the one with lowest proportion is shown here. F corresponds to failure.

NIST test

Table: NIST Test Results - Comparison of FResCA and NOCAS with first key-IV pair

	FResCA		NOCAS	
Test	P-val	Proportion	P-val	Proportion
Frequency	0.4944	98/100	0.9781	100/100
Block Frequency	0.5341	100/100	0.3505	99/100
Cumulative Sums*	0.2897	98/100	0.3669	100/100
Runs	0.7197	100/100	0.5544	100/100
Longest Run	0.2493	100/100	0.1917	97/100
Rank	0.2248	100/100	0.4012	100/100
FFT	0.6371	100/100	0.5544	100/100
NonOverlappingTemp.*	0.7792	97/100	F	F
OverlappingTemplate	0.7792	100/100	0.2248	99/100
Universal	0.3669	98/100	0.0712	99/100
Approximate Entropy	0.3345	99/100	0.4944	99/100
RandomExcursions*	0.7792	48/50	0.2133	64/68
RandomExcursionsVariant*	0.9835	49/50	0.5009	66/68
Serial*	0.9915	100/100	0.2757	99/100
LinearComplexity	0.2023	99/100	0.2493	100/100

*Note: In case of tests with more than one subset, the one with lowest proportion is shown here. F corresponds to failure.

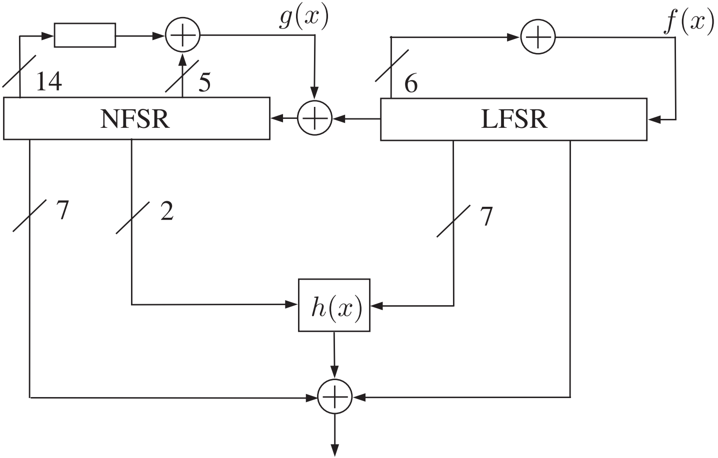
NIST test

Table: NIST Test Results - Comparison of FResCA and NOCAS with second key-IV pair

	FResCA		NOCAS	
Test	P-val	Proportion	P-val	Proportion
Frequency	0.7598	99/100	0.6787	97/100
Block Frequency	0.6163	99/100	0.1088	99/100
Cumulative Sums*	0.6993	99/100	0.2757	97/100
Runs	0.8343	100/100	0.7981	100/100
Longest Run	0.3669	100/100	0.0554	99/100
Rank	0.0146	99/100	0.6371	100/100
FFT	0.7598	100/100	0.6993	100/100
NonOverlappingTemp.*	0.0966	96/100	F	F
OverlappingTemplate	0.2622	99/100	0.4559	98/100
Universal	0.6282	98/100	0.7792	99/100
Approximate Entropy	0.4944	99/100	0.1917	100/100
RandomExcursions*	0.3345	56/57	0.8043	62/64
RandomExcursionsVariant*	0.2248	56/57	0.2328	63/64
Serial*	0.8343	99/100	0.7792	96/100
LinearComplexity	0.0590	100/100	0.3041	99/100

*Note: In case of tests with more than one subset, the one with lowest proportion is shown here. F corresponds to failure.

Grain-128



Grain-128

LFSR - $(s_i, s_{i+1}, \cdots, s_{i+127})$
NFSR - $(b_i, b_{i+1}, \cdots, b_{i+127})$

$$
\begin{aligned}
s_{i+128} &= s_i \oplus s_{i+7} \oplus s_{i+38} \oplus s_{i+70} \oplus s_{i+81} \oplus s_{i+96}. \\
b_{i+128} &= s_i \oplus b_i \oplus b_{i+26} \oplus b_{i+56} \oplus b_{i+91} \oplus b_{i+96} \\
&\quad \oplus b_{i+3}b_{i+67} \oplus b_{i+11}b_{i+13} \oplus b_{i+17}b_{i+18} \\
&\quad \oplus b_{i+27}b_{i+59} \oplus b_{i+40}b_{i+48} \oplus b_{i+61}b_{i+65} \\
&\quad \oplus b_{i+68}b_{i+84}.
\end{aligned}
$$

$$
\begin{aligned}
h &= b_{i+12}s_{i+8} \oplus s_{i+13}s_{i+20} \oplus b_{i+95}s_{i+42} \\
&\quad \oplus s_{i+60}s_{i+79} \oplus b_{i+12}b_{i+95}s_{i+95}. \\
z_i &= b_{i+2} \oplus b_{i+15} \oplus b_{i+36} \oplus b_{i+45} \oplus b_{i+64} \\
&\quad \oplus b_{i+73} \oplus b_{i+89} \oplus h \oplus s_{i+93}.
\end{aligned}
$$

Injecting Fault into Linear Block of Grain [3]

Attack Description

- finds out fault location by analysing keystream difference bits
- output z_i contains $s_{i+13}s_{i+20}$ and $s_{i+60}s_{i+79}$ terms. If fault injected at position 60, output difference gives the value of s_{79}. Each LFSR bit revealed in this way.
- Generates linear equations involving NFSR bits from the regular keystream.
- Grain reversible, runs backward to reveal the key.

Injecting Fault into Linear Block of Grain (continued)

Prevention in FResCA

- fault position determination fails. A single 1 in the register generates more 1's in different cell positions with each iteration
- In Grain, number of LFSR bits recovered depends on fault location and number of iterations after the fault injection
- In FResCA, fault gets dissipated to more and more neighbours with each iteration
- In Grain, more iterations produce more linear equations
- In FResCA, more iterations not fruitful as fault start mixing with more and more neighbours
- combination of nonlinear and linear CA rules makes FResCA non-reversible

Injecting Fault into Nonlinear Block of Grain[4]

Attack Description

- For a large number of key-IV pairs, b-bits provide unique keystream difference sequence for a particular fault position thereby revealing the fault position
- Fault Traces Table contains the list of corrupted bit locations after t iterations on fault injection at location f
- Equation for b_{128} contains seven terms of the form $b_m b_n$. Fault in m will reveal b_n.Linear b-terms in z_i gives more linear equations
- to determine LFSR bits, $b_{i+12}s_{i+8}$, $b_{i+95}s_{i+42}$, $b_{i+12}b_{i+95}s_{i+95}$ in z_i are exploited
- Grain run backwards to get the key

Injecting Fault into Nonlinear Block of Grain (continued)

Prevention in FResCA

- Fault propagation depends on the nonlinear CA rule and is propagated to neighbouring cells. Fast diffusion prevents unique pattern corresponding to a fault location. Computed fault traces look random
- No feedback path. Equation corresponding to b_{128} absent.
- In Grain, 7 single b-bit output taps - $b_2, b_{15}, b_{36}, b_{45}, b_{64}, b_{73}, b_{89}$. But only one b_{128} in FResCA.
- we cannot move a fault into single bit output tap
- Fault Traces Table cannot be created
- to find LFSR bits, fault propagated to specific positions $(b_m s_n)$ of NFSR without corrupting other b bits of z_i in Grain which is not possible in FResCA.
- FResCA not reversible

Summary

- designed a fault-resistant 4-neighbourhood CA based Grain-like cipher
- Cipher initialisation 8 times faster than Grain
- strong against different attacks, in particular, fault attacks
- experimental results show cipher's robustness

References

- The Estream Project, accessed 30 August 2016, http://www.ecrypt.eu.org/stream/
- Jose, J., Das, S., RoyChoudhury, D.:Inapplicability of fault attacks against trivium on a cellular automata based stream cipher. In: ACRI 2014. LNCS, vol. 8751, pp. 427–436, Springer (2014)
- Berzati, A., Canovas, C., Castagnos, G., Debraize, B., Goubin, L., Gouget, A., Paillier, P., Saldago, S.: Fault Analysis of Grain-128. In: HOST '09. pp. 7-14 (2009)
- Karmakar, S., RoyChowdhury, D.: Fault Analysis of Grain-128 by Targeting NFSR. In: AFRICACRYPT 2011. pp. 298-315 (2011)
- Jose, J., Roy Chowdhury, D.: Investigating Four Neighbourhood Cellular Automata as Better Cryptographic Primitives. J. Discrete Mathematical Sciences and Cryptography, to be published in 2016
- Bhaumik, J., RoyChowdhury, D.:NMix: An Ideal Candidate for Key Mixing. In: SECRYPT 2009. pp. 285–288 (2009)
- The NIST Statistical Test Suite, accessed 30 August 2016, http://crsc.nist.gov/groups/ST/toolkit/rng/

Thank You