A Differential Fault Attack on Grain-128a using MACs

Subhadeep Banik, Subhamoy Maitra and Santanu Sarkar

Applied Statistics Unit
Indian Statistical Institute, Kolkata
s.banik_r@isical.ac.in

Grain Family of Stream Ciphers

Grain Family

- Proposed by Hell et al in 2005
- Part of E-stream's hardware portfolio
- Bit-oriented, Synchronous stream cipher
- The first version (v0) of the cipher was crypatanalysed
  1. A Distinguishing attack by Kiaei et. al (Ecrypt : 071).
  2. A State Recovery attack by Berbain et.al (FSE 2006).
- After this, the versions Grain v1, Grain 128, Grain 128a were proposed.

General Structure of the Grain Family

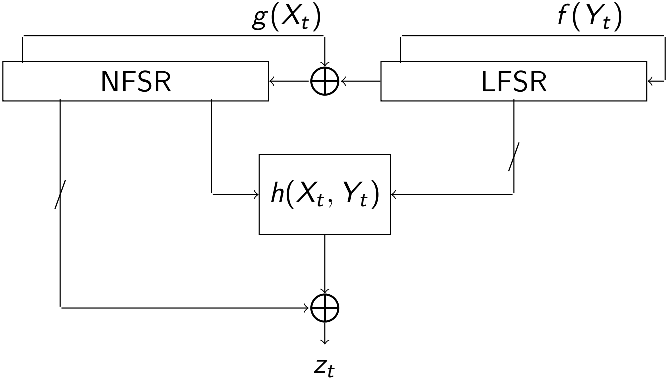

Figure: Structure of Grain v1

Grain-128a

The size of Key $n = 128$ bits and the IV is of size $m = 96$ bits. The value of pad used is $P = 0\text{x}\texttt{FFFF FFFE}$. The LFSR update rule is given by

$$y_{t+128} \overset{\Delta}{=} f(Y_t) = y_{t+96} + y_{t+81} + y_{t+70} + y_{t+38} + y_{t+7} + y_t.$$

The NFSR state is updated as follows

$$\begin{aligned} x_{t+128} = \; & y_t + g(x_{t+96}, x_{t+95}, x_{t+93}, x_{t+92}, x_{t+91}, x_{t+88}, x_{t+84}, x_{t+82}, \\ & x_{t+78}, x_{t+70}, x_{t+68}, x_{t+67}, x_{t+65}, x_{t+61}, x_{t+59}, x_{t+48}, \\ & x_{t+40}, x_{t+27}, x_{t+26}, x_{t+25}, x_{t+24}, x_{t+22}, x_{t+13}, x_{t+11}, \\ & x_{t+3}, x_t), \end{aligned}$$

where $g(x_{t+96}, x_{t+95}, \ldots, x_t)$ is defined as

$$\begin{aligned} g(X_t) = \; & x_t + x_{t+26} + x_{t+56} + x_{t+91} + x_{t+96} + x_{t+3}x_{t+67} + x_{t+11}x_{t+13} + \\ & x_{t+17}x_{t+18} + x_{t+27}x_{t+59} + x_{t+40}x_{t+48} + x_{t+61}x_{t+65} + x_{t+68}x_{t+84} + \\ & x_{t+88}x_{t+92}x_{t+93}x_{t+95} + x_{t+22}x_{t+24}x_{t+25} + x_{t+70}x_{t+78}x_{t+82}. \end{aligned}$$

Grain-128a

The keystream bit z_t is defined as

$$z_t = \bigoplus_{j \in A} x_{t+j} + y_{t+93} + h(x_{t+12}, y_{t+8}, y_{t+13}, y_{t+20}, x_{t+95}, y_{t+42}, y_{t+60}, y_{t+79}, y_{t+94})$$

where $A = \{2, 15, 36, 45, 64, 73, 89\}$ and

$$h(s_0, \ldots, s_8) = s_0 s_1 + s_2 s_3 + s_4 s_5 + s_6 s_7 + s_0 s_4 s_8 .$$

Keystream generating routines

- **Key Loading Algorithm (KLA)**
  - n-bit key $K \rightarrow$ NFSR
  - m-bit $(m < n)$ $IV \rightarrow$ LFSR[0]. . .LFSR[m-1]
  - $p = n - m$ bit pad $P \rightarrow$ LFSR[m]. . .LFSR[n-1]
- **Key Schedule Algorithm (KSA)**
  - For $2n$ clocks, output of h' is XOR-ed to the LFSR and NFSR update functions
  - $y_{t+n} = f(Y_t) + z_t$ and $x_{t+n} = y_t + z_t + g(X_t)$
- **Pseudo Random bitstream Generation Algorithm (PRGA)**
  - The feedback is discontinued
  - $y_{t+n} = f(Y_t)$ and $x_{t+n} = y_t + g(X_t)$
  - $z_t = h'(X^t, Y^t)$

Authentication procedure

- Message of length $L = m_0, \ldots, m_{L-1}$. Set $m_L = 1$ as padding.
- Use 2 registers: accumulator and shift register of 32 bits each.
- Initialize accumulator: $a_0^j = z_j, 0 \leq j \leq 31$
- Initialize Shift Register: $r_j = z_{32+j}, 0 \leq j \leq 31$.
- Update Shift Register: $r_{t+32} = z_{64+2t+1}$.
- Update Accumulator: $a_{t+1}^j = a_t^j + m_t r_{t+j}$ for $0 \leq j \leq 31$ and $0 \leq t \leq L$.
- The final content of accumulator, $a_{L+1}^0, \ldots, a_{L+1}^{31}$ is the output MAC.

Fault Attacks on Grain Family

- Fault Attack Grain-128 : Berzati et al. (IEEE HOST 2009),
- Fault Attack Grain-128 : Karmakar et. al. (Africacrypt 2011)
- Fault Attack on Grain v1, 128 : Banik et. al. (CHES 201)
- Fault Analysis of Grain-128a is tricky as the entire keystream is unavailable to the attacker
    - The first 64 and every alternate bit thereafter is used to compute MAC.

Fault Model

- The attacker is able to reset the system with the original Key-IV/ original Key and a different IV and start the cipher operations again.
- The attacker can inject a fault at any one random bit location of the LFSR.
- The fault in any bit may be reproduced at any later stage of operation, once injected.
- The attacker has full control over the timing of fault injection, i.e., it is possible to inject the fault precisely at any stage of the cipher operation.
- The attacker is able to obtain the original faulty MAC for any message of his choice.

Identifying Fault Location

Location Identification

- Apply a fault at a random LFSR location: imperative to determine fault location before proceeding.
- This is done by comparing the fault-free and faulty MACs of chosen messages.
- More than one fault at same location may be required to conclusively identify the location.

The Idea

- The MAC of the empty message $\sigma(\emptyset)$ is defined as

$$\sigma(\emptyset) = [z_0 + z_{32}, z_1 + z_{33}, \ldots, z_{31} + z_{63}],$$

and similarly the MAC for the single zero bit message is given by

$$\sigma(0) = [z_0 + z_{33}, z_1 + z_{34}, \ldots, z_{30} + z_{63}, z_{31} + z_{65}].$$

- Consider 2 initial states $S_0, S_{0,\Delta_{127}}$ such that $S_0 \oplus S_{0,\Delta_{127}} = y_{127}$

In all rounds $k \in [0, 65] \setminus \{33, 34, 48, 65\}$, the diifference does not affect output keystream bit.

$\Rightarrow$ Only 1st,2nd,16th bits of $\sigma(\emptyset)$ and $\sigma^{127}(\emptyset)$ may be different.

$\Rightarrow$ Only 0th,1st,15th,31st bits of $\sigma(0)$ and $\sigma^{127}(0)$ may be different.

Hence formulate signature vector Sgn_{127}= 9FFF 7FFF 3FFE FFFE.

- Idea is to match the sum of faultless and faulty keystream bits with all Sgn_ϕ for $\phi \in [0, 127]$

Beginning the Attack

Some Notations

- $S_t = [x_0^t, x_1^t, \ldots, x_{127}^t \ \ y_0^t, y_1^t, \ldots, y_{127}^t]$ state at round t of the PRGA. x_i^t $(y_i^t) \rightarrow i^{th}$ NFSR (LFSR) bit at t^{th} of the PRGA.
- When $t = 0$, $S_0 = [x_0, x_1, \ldots, x_{127} \ \ y_0, y_1, \ldots, y_{127}]$ for convenience.
- S_t^{ϕ} state round t of the PRGA, when a fault at LFSR location ϕ at round t.
- z_i^{ϕ} i^{th} faulty keystream bit, when a fault at LFSR location ϕ at round t.
- z_i is the fault-free i^{th} keystream bit.

Affine Differential Resistance

Definition

Consider a q-variable Boolean function F. A non-zero vector $\alpha \in \{0,1\}^q$ is said to be an affine differential of the function F if $F(\mathbf{x}) + F(\mathbf{x} + \alpha)$ is an affine function. A Boolean function is said to be affine differential resistant if it does not have any affine differential.

In Grain-128a

$$h(s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8) + h(s_0,s_1,1+s_2,s_3,s_4,s_5,s_6,s_7,s_8) = s_3 \quad (1)$$
$$h(s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8) + h(s_0,s_1,s_2,1+s_3,s_4,s_5,s_6,s_7,s_8) = s_2 \quad (2)$$
$$h(s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8) + h(s_0,s_1,s_2,s_3,s_4,s_5,1+s_6,s_7,s_8) = s_7 \quad (3)$$
$$h(s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8) + h(s_0,s_1,s_2,s_3,s_4,s_5,s_6,1+s_7,s_8) = s_6 \quad (4)$$

Therefore h is not affine differential resistant.

Determining the LFSR: An example

- Fault at LFSR location $\phi = 127$ at beginning of PRGA.
- At $t = 48$ differential travels to LFSR locn 79 $\Rightarrow S_{48}$ and S_{48}^{127} differ in locn 48 (corresponds to s_7 of h) and no other location of interest.
- By (4), $\Rightarrow z_{48} + z_{48}^{127} = h() + h(1 + s_7) = s_6 = y_{60}^{48} = y_{108}$.
- Also at $t = 16$, differential does not affect any location of interest $\Rightarrow z_{16} = z_{16}^{127}$.
- Now the sum of bit $d = 16$ of the original and faulty MACs of the null messsage is

$$\sigma(\emptyset) \oplus \sigma^{127}(\emptyset) = z_{16} + z_{48} + z_{16}^{127} + z_{48}^{127} = y_{108}.$$

- This gives one LFSR state bit of the initial PRGA state. By suitably varying ϕ, d 115 out of 128 bits can be recovered.

Finding the remaining LFSR bits

- Only State bits not found are $y_0, y_1, \ldots, y_{12}$.
- $\forall i \in [0, 12]$, Fault at $\phi = 109 + i \rightarrow$ sum of $(17 + i)^{th}$ bit of

$$\sigma(\emptyset) + \sigma^{109+i}(\emptyset) = y_{127}^{1+i}$$

- By LFSR update rule of Grain-128a we have

$$y_{127}^{1+i} = y_{96+i} + y_{81+i} + y_{70+i} + y_{38+i} + y_{7+i} + y_i, \quad \forall i \in [0, 12].$$

- In the last equation y_{12} is the only unknown: value calculated easily.
- Similarly y_{11} is the only unknown in the previous equation etc.

Determining the NFSR

We have, $h(\mathbf{s}) = s_0 \cdot u(\mathbf{s}) + v(\mathbf{s})$, where

$$u(\mathbf{s}) = s_1 + s_4 s_8, \;\; v(\mathbf{s}) = s_2 s_3 + s_4 s_5 + s_6 s_7$$

$$u(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) + u(s_0, 1 + s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) = 1, \tag{5}$$

$$v(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) + v(s_0, 1 + s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) = 0. \tag{6}$$

Also $h(\mathbf{s}) = s_4 \cdot U(\mathbf{s}) + V(\mathbf{s})$, where

$$U(\mathbf{s}) = s_5 + s_0 s_8, \;\; V(\mathbf{s}) = s_2 s_3 + s_4 s_5 + s_6 s_7$$

$$U(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) + U(s_0, s_1, s_2, s_3, s_4, 1 + s_5, s_6, s_7, s_8) = 1, \tag{7}$$

$$V(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8) + V(s_0, s_1, s_2, s_3, s_4, 1 + s_5, s_6, s_7, s_8) = 0. \tag{8}$$

Determining the NFSR: An example

- Fault at LFSR location $\phi = 8$ at beginning of PRGA.
- At $t = 0$ S_0 and S_0^8 differ in locn 8 (corresponds to s_1 of h) and no other location of interest.
- By (5,6), $\Rightarrow$
  $z_0 + z_0^8 = s_0 \cdot u() + v() + s_0 \cdot u(1 + s_1) + v(1 + s_1) = s_0 = x_{12}$.
- Also at $t = 32$, differential does not affect any location of interest $\Rightarrow z_{32} = z_{32}^8$.
- Now the sum of bit $d = 0$ of the original and faulty MACs of the null messsage is

$$\sigma(\emptyset) \oplus \sigma^8(\emptyset) = z_0 + z_{32} + z_0^8 + z_{32}^8 = x_{12}.$$

- This gives one NFSR state bit of the initial PRGA state. By suitably varying ϕ, d 97 out of 128 bits can be recovered.

Finding the remaining NFSR bits and Secret Key

- Only State bits not found are $x_0, x_1, \ldots, x_{11}, x_{76}, x_{77}, \ldots, x_{94}$.
- The remaining bits may be determined by a combination of solving equations and querying the device for MACs of other messages (Please refer to the paper).
- After the initial PRGA state is found we try to find the Secret Key.
- It is well known that the KSA routine of Grain is both one to one and Invertible.

$$S_0 \overset{KSA^{-1}}{\to} Secret\ Key$$

Thank You