Fault Attack on Grain Family of Stream Ciphers under Reasonable Assumptions

Subhadeep Banik, Subhamoy Maitra, Santanu Sarkar

s.banik_r@isical.ac.in
http://www.isical.ac.in/∼s.banik_r

December 10, 2012

Outline

1 Introduction

2 Fault model

3 Tools

4 The Attack

- Location Identification
- Determining the LFSR
- Determining the NFSR

5 Attacking the actual Ciphers

- Grain v1

6 Multiple bit fault

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

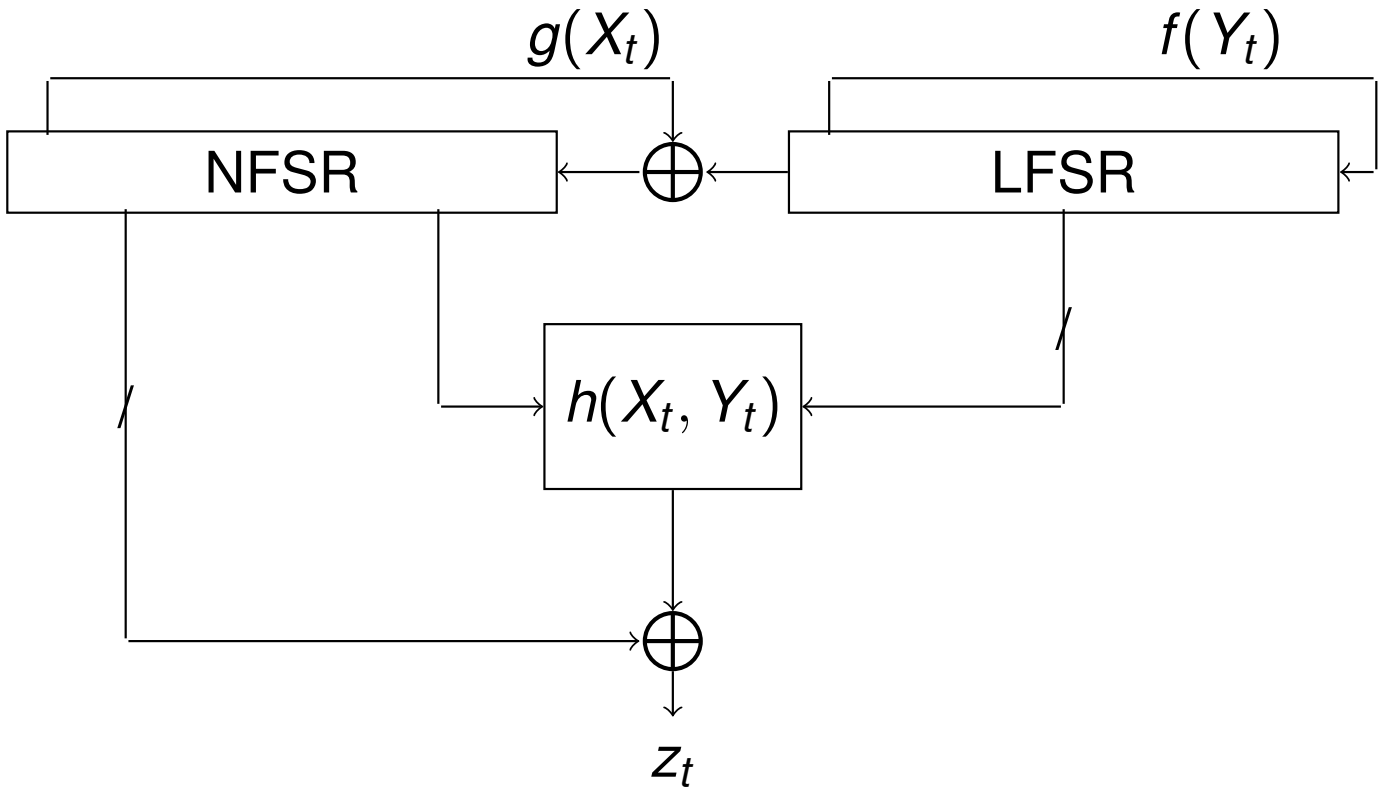


Figure: Structure of Grain family

Grain at a glance

	Grain v1	Grain-128	Grain-128a
n	80	128	128
m	64	96	96
Pad	FFFF	FFFFFFFF	FFFFFFFE
$f(\cdot)$	$y_{t+62} \oplus y_{t+51} \oplus y_{t+38} \oplus y_{t+23} \oplus y_{t+13} \oplus y_t$	$y_{t+96} \oplus y_{t+81} \oplus y_{t+70} \oplus y_{t+38} \oplus y_{t+7} \oplus y_t$	$y_{t+96} \oplus y_{t+81} \oplus y_{t+70} \oplus y_{t+38} \oplus y_{t+7} \oplus y_t$
$g(\cdot)$	$x_{t+62} \oplus x_{t+60} \oplus x_{t+52} \oplus x_{t+45} \oplus x_{t+37} \oplus x_{t+33} \oplus x_{t+28} \oplus x_{t+21} \oplus x_{t+14} \oplus x_{t+9} \oplus x_t \oplus x_{t+63}x_{t+60} \oplus x_{t+37}x_{t+33} \oplus x_{t+15}x_{t+9} \oplus x_{t+60}x_{t+52}x_{t+45} \oplus x_{t+33} \oplus x_{t+28}x_{t+21} \oplus x_{t+63}x_{t+60} \oplus x_{t+21}x_{t+15} \oplus x_{t+63}x_{t+60} \oplus x_{t+52}x_{t+45}x_{t+37} \oplus x_{t+33} \oplus x_{t+28}x_{t+21}x_{t+15}x_{t+9} \oplus x_{t+52}x_{t+45}x_{t+37}x_{t+33} \oplus x_{t+28}x_{t+21}$	$y_t \oplus x_t \oplus x_{t+26} \oplus x_{t+56} \oplus x_{t+91} \oplus x_{t+96} \oplus x_{t+3}x_{t+67} \oplus x_{t+11}x_{t+13} \oplus x_{t+17}x_{t+18} \oplus x_{t+27}x_{t+59} \oplus x_{t+40}x_{t+48} \oplus x_{t+61} \oplus x_{t+65} \oplus x_{t+68}x_{t+84}$	$y_t \oplus x_t \oplus x_{t+26} \oplus x_{t+56} \oplus x_{t+91} \oplus x_{t+96} \oplus x_{t+3}x_{t+67} \oplus x_{t+11}x_{t+13} \oplus x_{t+17}x_{t+18} \oplus x_{t+27}x_{t+59} \oplus x_{t+40}x_{t+48} \oplus x_{t+61} \oplus x_{t+65} \oplus x_{t+68}x_{t+84} \oplus x_{t+88}x_{t+92}x_{t+93}x_{t+95} \oplus x_{t+22}x_{t+24}x_{t+25} \oplus x_{t+70}x_{t+78}x_{t+82}$
$h(\cdot)$	$y_{t+3}y_{t+25}y_{t+46} \oplus y_{t+3}y_{t+46}y_{t+64} \oplus y_{t+3}y_{t+46} \oplus x_{t+63} \oplus y_{t+25}y_{t+46}x_{t+63} \oplus y_{t+46}y_{t+64}x_{t+63} \oplus y_{t+3}y_{t+64} \oplus y_{t+46}y_{t+64} \oplus y_{t+64} \oplus x_{t+63} \oplus y_{t+25} \oplus x_{t+63}$	$x_{t+12}x_{t+95}y_{t+95} \oplus x_{t+12}y_{t+8} \oplus y_{t+13}y_{t+20} \oplus x_{t+95}y_{t+42} \oplus y_{t+60}y_{t+79}$	$x_{t+12}x_{t+95}y_{t+94} \oplus x_{t+12}y_{t+8} \oplus y_{t+13}y_{t+20} \oplus x_{t+95}y_{t+42} \oplus y_{t+60}y_{t+79}$
z_t	$x_{t+1} \oplus x_{t+2} \oplus x_{t+4} \oplus x_{t+10} \oplus x_{t+31} \oplus x_{t+43} \oplus x_{t+56} \oplus h$	$x_{t+2} \oplus x_{t+15} \oplus x_{t+36} \oplus x_{t+45} \oplus x_{t+64} \oplus x_{t+73} \oplus x_{t+89} \oplus y_{t+93} \oplus h$	$x_{t+2} \oplus x_{t+15} \oplus x_{t+36} \oplus x_{t+45} \oplus x_{t+64} \oplus x_{t+73} \oplus x_{t+89} \oplus y_{t+93} \oplus h$

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 $x_{t+n} = y_t + g(X_t)$
  - $z_t = h'(X^t, Y^t)$

A survey of Fault attacks on Grain

	HOST'08	Africacrypt'10	CHES'12	This work
Multiple fault at same location	Required	Required	Required	**Not Required**
Multiple IV Initialization	No	Yes	No	No
Single bit Fault	Yes	Yes	Yes	**Upto 3-bit toggle allowed**
Control over Fault Timing	Required	Required	Required	Required
Multiple Re-Keying	Allowed	Allowed	Allowed	Allowed

Figure: Differential Fault Attack on Grain: Survey of Fault Models.

Fault Model

- The attacker can reset the cipher with the original Key-IV and restart cipher operations as many times he wishes.
- He has full control over the timing of fault injection.
- He can inject a fault that may affect upto k consecutive LFSR locations but he is unaware of the exact number of bits altered. In this work we have concentrated on the case when $k = 3$.

Differential Grain

The engine Δ_ϕ-Grain

Generalised Grain

- n-bit LFSR and NFSR
- LFSR update

$$y_{t+n} = y_t \oplus y_{t+f_1} \oplus y_{t+f_2} \oplus \cdots \oplus y_{t+f_a}$$

- NFSR updates as

$$x_{t+n} = y_t \oplus g(x_t, x_{t+g_1}, x_{t+g_2}, \ldots, x_{t+g_b})$$

Here OR is map from $\mathbb{Z}^{b+1} \rightarrow \{0, 1\}$

Δ_ϕ-Grain

- n-integer LFSR and NFSR
- LFSR updates as

$$u_{t+n} = u_t + u_{t+f_1} + \cdots + u_{t+f_a} \bmod 2$$

- NFSR updates as

$$v_{t+n} = u_t + 2 \cdot OR(v_t, v_{t+g_1}, \ldots, v_{t+g_b})$$

$$OR(k_0, k_1, \ldots, k_b) = \begin{cases} 0, & \text{if } k_0 = k_1 = k_2 = \cdots = k_b = 0, \\ 1, & \text{otherwise.} \end{cases}$$

- Generalised Grain: Key stream bit

$$z_t = x_{t+l_1} \oplus x_{t+l_2} \oplus \cdots \oplus x_{t+l_c} \oplus y_{t+i_1} \oplus y_{t+i_2} \oplus \cdots \oplus y_{t+i_d} \oplus h(y_{t+h_1}, y_{t+h_2}, \ldots, y_{t+h_e}, x_{t+j_1}, x_{t+j_2}, \ldots, x_{t+j_w})$$

- Let $\chi_t = [v_{t+l_1}, v_{t+l_2}, \ldots, v_{t+l_c}, u_{t+i_1}, u_{t+i_2}, \ldots, u_{t+i_d}]$
  and $\Upsilon_t = [u_{t+h_1}, u_{t+h_2}, \ldots, u_{t+h_e}, v_{t+j_1}, v_{t+j_2}, \ldots, v_{t+j_w}]$
- Δ_ϕ-Grain: Output Element

$$\Delta z_t = \begin{cases} 0, & \text{if } \Upsilon_t = \mathbf{0} \text{ AND } \chi_t \sqsubseteq 1 \text{ AND } |\chi_t| \text{ is even} \\ 1, & \text{if } \Upsilon_t = \mathbf{0} \text{ AND } \chi_t \sqsubseteq 1 \text{ AND } |\chi_t| \text{ is odd} \\ 2, & \text{otherwise.} \end{cases}$$

Note $\mathbf{V} \sqsubseteq \beta \Rightarrow$ all elements of $\mathbf{V}$ are less than or equal to β

Differential Grain

- $S_t \Rightarrow t^{th}$ state of the generalised Grain cipher.
- $S_{t,\Delta_\phi} \Rightarrow t^{th}$ state when faulted at LFSR location ϕ at start of PRGA.
- $\Delta L, \Delta N$ are initialized to all zero except the ϕ^{th} location of ΔL which is set to 1.

Interpretation of the cell values

- Some cell in ΔL_t or ΔN_t is 0, $\Rightarrow$ corresponding bits in S_t and S_{t,Δ_ϕ} are equal with probability 1
- Some cell in ΔL_t or ΔN_t is 1, $\Rightarrow$ corresponding bits in S_t and S_{t,Δ_ϕ} are unequal with probability 1
- Some cell in ΔL_t or ΔN_t is 2/3, $\Rightarrow$ the corresponding bits in S_t and S_{t,Δ_ϕ} are different with some probability $0 < p_d < 1$.

Differential Grain

Interpretation of the output elements

- $\Delta z_t = 0 \Rightarrow$ corresponding keystream bit produced by S_t and S_{t,Δ_ϕ} are equal with probability 1
- $\Delta z_t = 1 \Rightarrow$ corresponding keystream bit produced by S_t and S_{t,Δ_ϕ} are unequal with probability 1
- $\Delta z_t = 2/3 \Rightarrow$ corresponding keystream bit produced by S_t and S_{t,Δ_ϕ} are equal with some probability $0 < p_d < 1$.

Differential Grain

The algorithm D-GRAIN(ϕ, r)

Input: ϕ: An LFSR location $\in [0, n-1]$, an integer $r(> 0)$;
Output: An integer array ΔZ of r elements;
Output: Two integer arrays χ_t, Υ_t for $0 \leq t < r$;

$[u_0, u_1, \ldots, u_{n-1}] \leftarrow \mathbf{0}, [v_0, v_1, \ldots, v_{n-1}] \leftarrow \mathbf{0}, u_\phi \leftarrow 1, t \leftarrow 0;$
while $t < r$ **do**
  $\Upsilon_t \leftarrow [u_{h_1}, u_{h_2}, \ldots, u_{h_e}, v_{j_1}, v_{j_2}, \ldots, v_{j_w}]$;
  $\chi_t \leftarrow [v_{l_1}, v_{l_2}, \ldots, v_{l_c}, u_{i_1}, u_{i_2}, \ldots, u_{i_d}];$
  if $\Upsilon_t = \mathbf{0}$ *AND* $\chi_t \sqsubseteq 1$ **then**
    if $|\chi_t|$ *is EVEN* **then**
      $\Delta z_t \leftarrow 0;$
    end
    if $|\chi_t|$ *is ODD* **then**
      $\Delta z_t \leftarrow 1;$
    end
  end
  else
    $\Delta z_t \leftarrow 2;$
  end
  $t_1 \leftarrow u_0 + u_{f_1} + u_{f_2} + \ldots + u_{f_a} \bmod 2;$
  $t_2 \leftarrow u_0 + 2 \cdot OR(v_0, v_{g_1}, v_{g_2}, \ldots, v_{g_b});$
  $[u_0, u_1, \ldots, u_{n-2}, u_{n-1}] \leftarrow [u_1, u_2, \ldots, u_{n-1}, t_1];$
  $[v_0, v_1, \ldots, v_{n-2}, v_{n-1}] \leftarrow [v_1, v_2, \ldots, v_{n-1}, t_2];$
  $t = t + 1;$
end
$\Delta Z = [\Delta z_0, \Delta z_1, \ldots, \Delta z_{r-1}];$

Return $[\chi_0, \chi_1, \ldots, \chi_{r-1}]$, $[\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_{r-1}]$, ΔZ

Beginning the Attack

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 Key-streams.
- More than one fault at same location is no longer required to conclusively identify the location.

The Idea

- Consider 2 initial states in Grain v1 $S_0, S_{0,\Delta_{79}}$ such that $S_0 \oplus S_{0,\Delta_{79}} = s_{79}$
- In all rounds $k \in [0, 79] \setminus \{15, 33, 44, 51, 54, 57, 62, 69, 72, 73, 75, 76\}$, the difference does not affect output keystream bit.
At all these rounds output of $S_0, S_{0,\Delta_{79}}$ guaranteed to be equal. Hence formulate first signature vector Sgn^1_{79}= FFFE FFFF BFF7 EDBD FB27...
- In round $k = 103$, the difference only affects one linear mask bit.
At this round output of $S_0, S_{0,\Delta_{79}}$ guaranteed to be unequal. Hence formulate second signature vector Sgn^2_{79}= 0000 0000 0000 0000 0000 0000 0100...
- Idea is to match the sum of faultless and faulty keystream bits with all Sgn^1_ϕ, Sgn^2_ϕ for $\phi \in [0, 79]$

How to construct signature vectors

The idea is to use the routine D-GRAIN(ϕ, r). It is easy to see

$$Sgn^1_\phi(i) = \begin{cases} 1, & \text{if } \Delta_\phi z_i = 0, \\ 0, & \text{otherwise.} \end{cases} \qquad Sgn^2_\phi(i) = \begin{cases} 1, & \text{if } \Delta_\phi z_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Attack Procedure

A. Reset the cipher with the unknown key K and Initial Vector IV and record the first $2n$ fault-free key-stream bits Z.

B. Reset the cipher with K, IV, Inject a single bit fault in a random LFSR location ϕ, $0 \leq \phi \leq n-1$ at the beginning of the PRGA. Record the faulty key-stream bits Z^{ϕ}.

C. Repeat Step **B** around $n \ln n$ times so that n different faulty key-stream vectors corresponding to all LFSR locations $\phi \in [\,0, n-1]$ are obtained.

D. Once all the faulty key-stream vectors have been labeled we proceed to the next stage of the attack.

Some Notations

- $S_t = [x_0^t, x_1^t, \ldots, x_{n-1}^t \;\; y_0^t, y_1^t, \ldots, y_{n-1}^t] \Rightarrow t^{th}$ internal state.
- For convenience $S_0 = [x_0, x_1, \ldots, x_{n-1} \;\; y_0, y_1, \ldots, y_{n-1}$.
- $S_{t,\Delta_\phi} \Rightarrow t^{th}$ internal state when LFSR location ϕ is faulted at the start of PRGA.
- $z_i^\phi \Rightarrow i^{th}$ key-stream bit, after LFSR location ϕ is faulted at the start of PRGA. z_i is the fault-free i^{th} key-stream bit.
- $\eta_t = [x_{l_1}^t, x_{l_2}^t \ldots, x_{l_c}^t, y_{i_1}^t, y_{i_2}^t \ldots, y_{i_d}^t]$ is the subset of S_t which forms the linear mask and $\theta_t = [y_{h_1}^t, y_{h_2}^t, \ldots, y_{h_e}^t, x_{j_1}^t, x_{j_2}^t, \ldots, x_{j_w}^t]$ be the subset of S_t which forms the input to h.
- $\widetilde{\mathbf{v}} \Rightarrow \mathbf{v}$ in GF(2).
- $\mathcal{P}(\mathbf{w})$ denotes the GF(2) sum of the elements of $\mathbf{w}$.

Determining the LFSR

During PRGA, the LFSR evolves linearly and independent of the NFSR.

$\Rightarrow$ y_i^t for any $i \in [0, n-1]$ and $t \geq 0$ is a linear function of $y_0, y_1, \ldots, y_{n-1}$ (Initial LFSR State).

$\Rightarrow$ We will use the algorithm D-GRAIN(ϕ, r) to get n linear equations on $y_0, y_1, \ldots, y_{n-1}$.

$\Rightarrow$ $[\chi_{0,\phi}, \chi_{1,\phi}, \ldots, \chi_{2n-1,\phi}]$, $[\Upsilon_{0,\phi}, \Upsilon_{1,\phi}, \ldots, \Upsilon_{2n-1,\phi}]$, $\Delta_\phi Z$ be the outputs of D-GRAIN$(\phi, 2n)$.

A Definition

Definition

Consider a q-variable Boolean function F.

$$\hat{F}(\mathbf{x}) = c_1 F(\mathbf{x} \oplus \alpha_1) \oplus c_2 F(\mathbf{x} \oplus \alpha_2) \oplus \cdots \oplus c_i F(\mathbf{x} \oplus \alpha_i),$$

where $c_1, c_2, \ldots, c_i \in \{0, 1\}$ is said to be a derivative of F.
If $\hat{F}$ is an **affine** Boolean function and $c_1 = \cdots = c_i = 1$ then

$$\pi = [\alpha_1, \alpha_2, \ldots, \alpha_i]$$

is **an affine differential tuple**(ADT) of F.
$\Rightarrow$ If $\alpha_i \neq \mathbf{0}$ $\forall i$ then π is weight i ADT.
$\Rightarrow$ Else π is Weight $i - 1$ ADT.

Obtaining and Using the ADTs

λ is odd

$\Rightarrow$ Consider λ fault locations $\phi_i \in [0, n-1]$.

$\Rightarrow$ Let $[\chi_{0,\phi_i}, \ldots, \chi_{2n-1,\phi_i}]$, $[\Upsilon_{0,\phi_i}, \ldots, \Upsilon_{2n-1,\phi_i}]$, $\Delta_{\phi_i} Z$ be the λ outputs of D-GRAIN$(\phi_i, 2n)$ for $i \in [1, \lambda]$.

$\Rightarrow$ Let $[\mathbf{0}, \alpha_1, \alpha_2, \ldots, \ldots, \alpha_\lambda]$ be a weight λ (λ odd) ADT of h,

$$h(\mathbf{x}) \oplus \bigoplus_{i=1}^{\lambda} h(\mathbf{x} \oplus \alpha_i) = H_1(\mathbf{x})$$

is a function of **LFSR** variables only.

$\Rightarrow$ If for some t, $\chi_{t,\phi_i} \sqsubseteq 1$, $\Upsilon_{t,\phi_i} \sqsubseteq 1$ and $\widetilde{\Upsilon}_{t,\phi_i} = \alpha_i$ for all $i \in [1, \lambda]$, then

$$\begin{aligned} z_t \oplus \bigoplus_{i=1}^{\lambda} z_t^{\phi_i} &= \mathcal{P}(\eta_t) \oplus h(\theta_t) \oplus \bigoplus_{i=1}^{\lambda} \left(\mathcal{P}(\eta_t \oplus \widetilde{\chi}_{t,\phi_i}) \oplus h(\theta_t \oplus \widetilde{\Upsilon}_{t,\phi_i}) \right) \\ &= \bigoplus_{i=1}^{\lambda} \mathcal{P}(\widetilde{\chi}_{t,\phi_i}) \oplus H_1(\theta_t). \end{aligned}$$

$\Rightarrow$ One linear equation in $y_0, y_1, \ldots, y_{n-1}$!!!!

Obtaining and Using the ADTs

λ is even

$\Rightarrow$ Let $[\alpha_1, \alpha_2, \ldots, \ldots, \alpha_\lambda]$ be a weight λ (λ even) ADT of h,

$$\bigoplus_{i=1}^{\lambda} h(\mathbf{x} \oplus \alpha_i) = H_2(\mathbf{x})$$

is a function of **LFSR** variables only.

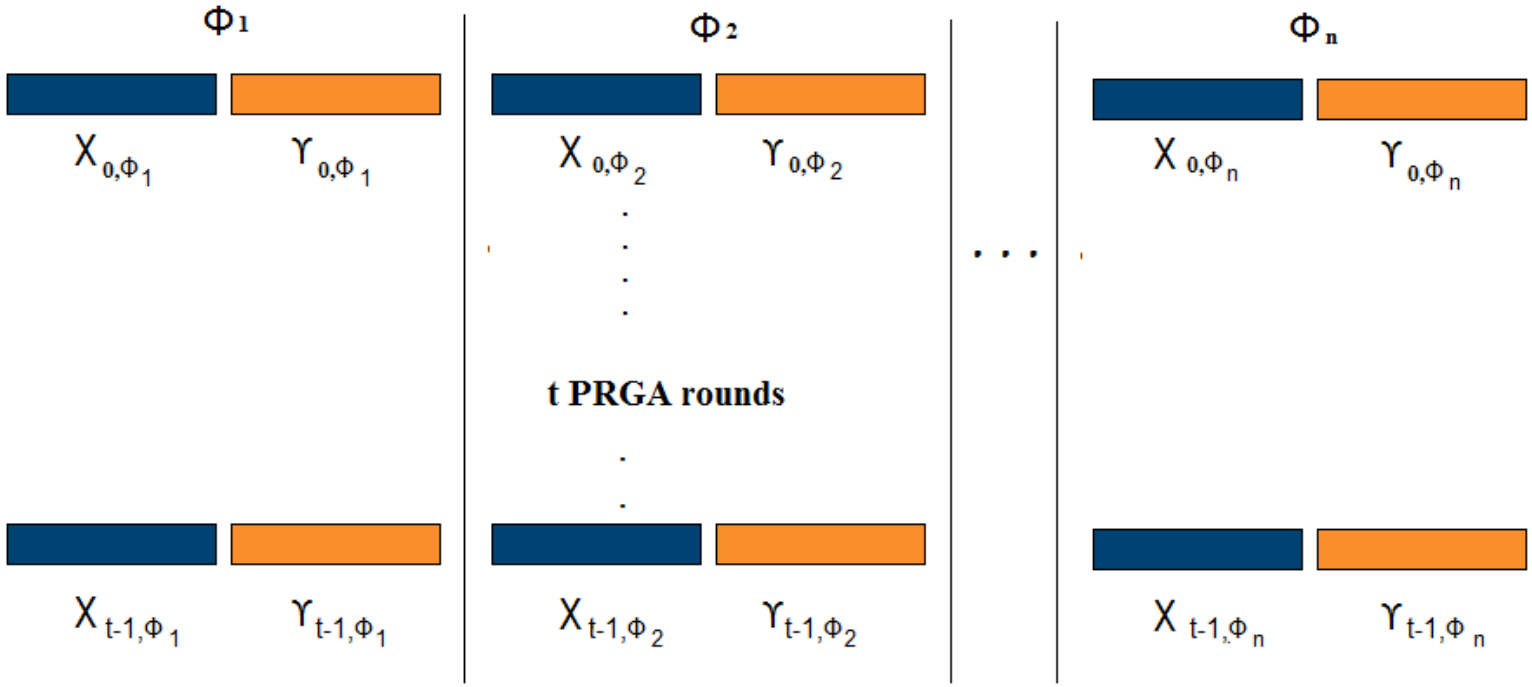
$\Rightarrow$ If for some t, $\chi_{t,\phi_i} \sqsubseteq 1$, $\Upsilon_{t,\phi_i} \sqsubseteq 1$ and $\widetilde{\Upsilon}_{t,\phi_i} = \alpha_i$ for all $i \in [1, \lambda]$, then

$$\begin{aligned} \bigoplus_{i=1}^{\lambda} z_t^{\phi_i} &= \bigoplus_{i=1}^{\lambda} \left(\mathcal{P}(\eta_t \oplus \widetilde{\chi}_{t,\phi_i}) \oplus h(\theta_t \oplus \widetilde{\Upsilon}_{t,\phi_i}) \right) \\ &= \bigoplus_{i=1}^{\lambda} \mathcal{P}(\widetilde{\chi}_{t,\phi_i}) \oplus H_2(\theta_t). \end{aligned}$$

$\Rightarrow$ One linear equation in $y_0, y_1, \ldots, y_{n-1}$!!!!

$\Rightarrow$ Varying over λ one can get n linearly independent equations on $y_0, y_1, \ldots, y_{n-1}$

The algorithm $\mathrm{FLE}_L(\lambda)$



Total search complexity : $t \times \binom{n}{\lambda}$

Determining the NFSR

$\Rightarrow$ Let $[\alpha_1, \alpha_2, \ldots, \ldots, \alpha_\lambda]$ be a weight λ (λ odd) ADT of h,

$$h(\mathbf{x}) + \bigoplus_{i=1}^{\lambda} h(\mathbf{x} \oplus \alpha_i) = H_1(\mathbf{x}) = x' \oplus H_{11}(\mathbf{x})$$

where x' is a variable that takes input from an NFSR location and $H_{11}(\mathbf{x})$ is a function only on the **LFSR** variables.

$\Rightarrow$ If for some t, $\chi_{t,\phi_i} \sqsubseteq 1$, $\Upsilon_{t,\phi_i} \sqsubseteq 1$ and $\widetilde{\Upsilon}_{t,\phi_i} = \alpha_i$ for all $i \in [1, \lambda]$, then

$$\begin{aligned} z_t \oplus \bigoplus_{i=1}^{\lambda} z_t^{\phi_i} &= \mathcal{P}(\eta_t) \oplus h(\theta_t) \oplus \bigoplus_{i=1}^{\lambda} \left(\mathcal{P}(\eta_t \oplus \widetilde{\chi}_{t,\phi_i}) \oplus h(\theta_t \oplus \widetilde{\Upsilon}_{t,\phi_i}) \right) \\ &= \bigoplus_{i=1}^{\lambda} \mathcal{P}(\widetilde{\chi}_{t,\phi_i}) \oplus H_1(\theta_t) = \bigoplus_{i=1}^{\lambda} \mathcal{P}(\widetilde{\chi}_{t,\phi_i}) \oplus H_{11}(\theta_t) \oplus x_{j_r}^t \end{aligned}$$

$\Rightarrow$ One NFSR state bit x_j^t is obtained!!!!

$\Rightarrow$ Varying over λ one can get n NFSR state bits for some PRGA round t.

The Algorithm $\text{FLE}_N(\lambda)$

- Same as $\text{FLE}_L(\lambda)$.
- Search for differential tuples that will lead to functions of the form $x' \oplus H_{11}(\mathbf{x})$.
- Complexity same as $\text{FLE}_L(\lambda) = t \times \binom{n}{\lambda}$

Getting the secret Key

- At this time we have
  1. All LFSR state bits of S_0.
  2. All NFSR state bits of S_t for some t.
- Calculate LFSR state bits of S_t by running the PRGA t times.
- It is well known that in Grain, the state updates in KSA and PRGA are one-to-one and invertible.

$$S_t \overset{PRGA^{-1}}{\rightarrow} S_0 \overset{KSA^{-1}}{\rightarrow} \text{Secret Key}$$

Grain v1

t	ϕ_1	ϕ_2	ϕ_3	Range	Expr.	ADT
$45+i$	$62+i$	$24+i$	$70+i$	$i \in [0,9]$	y_{46}^t	00000, 00100, 00110, 01000
$55+i$	$72+i$	$16+i$	$51+i$	$i \in [0,7]$		
$63+i$	$13+i$	$24+i$	$59+i$	$i \in [0,9]$		
$73+i$	$33+i$	$26+i$	$51+i$	$i \in [0,10]$		
$84+i$	$44+i$	$37+i$	$38+i$	$i \in [0,6]$		
$91+i$	$53+i$	$44+i$	$41+i$	$i \in [0,8]$		
$100+i$	$70+i$	$53+i$	$60+i$	$i \in [0,8]$		
109	79	71	69			
$77+i$	$45+i$	$51+i$	$38+i$	$i \in [0,5]$	$y_3^t \oplus y_{25}^t \oplus y_{64}^t$	00000, 01100, 10000, 10110
$83+i$	$72+i$	$57+i$	$44+i$	$i \in [0,4]$		
94	62	79	55			
95	78	63	56		$y_3^t \oplus y_{25}^t \oplus y_{46}^t \oplus y_{64}^t$	00000, 01001, 01100, 10110

Table: Output of $\mathrm{FLE}_L(3)$ for Grain v1 (ADT implies Affine Differential Tuple)

Grain v1

t	ϕ_1	ϕ_2	Range	Expr.	ADT
$110 + i$	$64 + i$	$77 + i$	$i \in [0, 1]$	y_{46}^t	00001, 11000

Table: Output of $\text{FLE}_L(2)$ for Grain v1

t	ϕ_1	Range	Expr.	ADT
$55 + i$	$23 + i$	$i \in [0, 14]$	$1 \oplus y_3^t \oplus y_{46}^t \oplus x_{63}^t$	00000, 01010
$70 + i$	$77 + i$	$i \in [0, 2]$		
$91 + i$	$62 + i$	$i \in [0, 5]$		

Table: Output of $\text{FLE}_N(1)$ for Grain v1

t	ϕ_1	ϕ_2	ϕ_3	Range	Expr.	ADT
$17 + i$	i	$1 + i$	$20 + i$	$i \in [0, 27]$	$1 \oplus y_3^t \oplus y_{46}^t \oplus x_{63}^t$	00000, 00001, 00010, 10000
$45 + i$	$28 + i$	$13 + i$	$48 + i$	$i \in [0, 9]$		
$73 + i$	$53 + i$	$33 + i$	$26 + i$	$i \in [0, 17]$	$1 \oplus y_3^t \oplus x_{63}^t$	00000, 00010, 00100, 00110

Table: Output of $\text{FLE}_N(3)$ for Grain v1

Multiple bit fault

- This work considers the case when fault affects 3 contiguous LFSR locations.
- Four cases possible
  1. Exactly one LFSR bit is flipped (n cases),
  2. 2 consecutive locations $i, i+1$ of the LFSR are flipped ($n-1$ cases),
  3. 3 consecutive locations $i, i+1, i+2$ of the LFSR are flipped ($n-2$ cases) and
  4. Locations $i, i+2$ are flipped but not $i+1$ ($n-2$ cases).
- The attack is still possible with high probability.
- Assuming each of the cases is possible with equal probability the number of rekeyings increases to $(4n-5)\ln(4n-5)$.

The Idea

- Fault location algorithm: If a faulty keystream is due to a multiple bit fault then the algorithm returns error.
- If faulty keystream is due to a single bit fault the algorithm returns the required location.
- The probability of the above is
  - 0.99994 for Grain v1,
  - 1.00 for Grain-128 and
  - 0.993 for Grain-128a.
- Probabilities verified by computer simulations over 2^{20} Key-IV pairs.

THANK YOU