Fast Correlation Attacks on Grain-like Small State Stream Ciphers

Bin Zhang*, Xinxin Gong* and Willi Meier**

* Chinese Academy of Sciences
**FHNW,Switzerland

Outline

Background

- As a rule of thumb, the internal state size of modern stream ciphers is at least twice as large as the key size, as seen from the European eSTREAM project.
  - Grain v1, 160-bit internal state $+$ 160 initialization rounds $\rightarrow$ 80-bit security
  - Trivium, 288-bit internal state $+$ 1152 initialization rounds $\rightarrow$ 80-bit security
- On the other hand, the most power consuming component is the number of memory gates, corresponding to the internal state size of the primitive.
- How about other design paradigm ?

Small State Stream Ciphers (1)

At FSE 2015,

- Another design paradigm is proposed and instantiated by a new design, called Sprout.

  Property 1: the size of the internal state is reduced, and thus the hardware area.

  Property 2: the non-linear state updating is dependent on the secret key.

- A key-dependent state update, in both initialization and keystream generation phases, to resist the classical TMD tradeoff attacks.

- NFSR-based mechanisms to thwart (fast) correlation attacks and algebraic attacks.

Small State Stream Ciphers (1)

At FSE 2015,

- Another design paradigm is proposed and instantiated by a new design, called Sprout.

  Property 1: the size of the internal state is reduced, and thus the hardware area.

  Property 2: the non-linear state updating is dependent on the secret key.

- A key-dependent state update, in both initialization and keystream generation phases, to resist the classical TMD tradeoff attacks.

- NFSR-based mechanisms to thwart (fast) correlation attacks and algebraic attacks.

Small State Stream Ciphers (1)

At FSE 2015,

- Another design paradigm is proposed and instantiated by a new design, called Sprout.

  Property 1: the size of the internal state is reduced, and thus the hardware area.

  Property 2: the non-linear state updating is dependent on the secret key.

- A key-dependent state update, in both initialization and keystream generation phases, to resist the classical TMD tradeoff attacks.
- NFSR-based mechanisms to thwart (fast) correlation attacks and algebraic attacks.

Small State Stream Ciphers (1)

At FSE 2015,

- Another design paradigm is proposed and instantiated by a new design, called Sprout.

  Property 1: the size of the internal state is reduced, and thus the hardware area.
  Property 2: the non-linear state updating is dependent on the secret key.

- A key-dependent state update, in both initialization and keystream generation phases, to resist the classical TMD tradeoff attacks.

- NFSR-based mechanisms to thwart (fast) correlation attacks and algebraic attacks.

Small State Stream Ciphers (1)

At FSE 2015,

- Another design paradigm is proposed and instantiated by a new design, called Sprout.

  Property 1: the size of the internal state is reduced, and thus the hardware area.

  Property 2: the non-linear state updating is dependent on the secret key.

- A key-dependent state update, in both initialization and keystream generation phases, to resist the classical TMD tradeoff attacks.

- NFSR-based mechanisms to thwart (fast) correlation attacks and algebraic attacks.

Cryptanalysis of Sprout

- Many attacks on Sprout appeared after FSE 2015.
  - V. Lallemand and M. Naya-Plasencia, Cryptanalysis of full Sprout, Crypto 2015
  - Esgin M. F. and Kara O., Practical cryptanalysis of full Sprout with TMD tradeoff attack, SAC 2015
  - B. Zhang. and X. Gong., Another tradeoff attack on Sprout-like stream ciphers, Asiacrypt 2015
- To remedy the situation, Fruit, Plantlet and Lizard are proposed.

Small State Stream Ciphers (2)

Cryptanalysis of Sprout

- Many attacks on Sprout appeared after FSE 2015.
  - V. Lallemand and M. Naya-Plasencia, Cryptanalysis of full Sprout, Crypto 2015
  - Esgin M. F. and Kara O., Practical cryptanalysis of full Sprout with TMD tradeoff attack, SAC 2015
  - B. Zhang. and X. Gong., Another tradeoff attack on Sprout-like stream ciphers, Asiacrypt 2015
- To remedy the situation, Fruit, Plantlet and Lizard are proposed.

Small State Stream Ciphers (3)

- The lack of a well-understood theoretical study in this domain apparently restricts the confidence that people have on such primitives.
- It is expected that lower area, thus power consumption could be achieved by using a fixed non-volatile secret key and the key -dependent state updating in an adequate way.
- This motivates us to study the security of these small primitives against a new type of attacks that is well-tailored for them.

Our Contributions

Study the security of these Grain-like small state stream ciphers by fast correlation attacks, the classical cryptanalytic methods against LFSR -based stream ciphers.

- Define a generalized model, which adopts a cascaded structure to connect several NFSRs and exploits the key-dependent state updating in the keystream generation phase.
- It is shown that if the non-linear combining function used to generate the final keystream has some pseudo-linear properties, we could restore the full internal state of the model in a divide-and-conquer manner.
- For Fruit, it requires $2^{62.8}$ Fruit encryptions and $2^{22.3}$ keystream bits for all the 80-bit secret keys, verified by experiments on a small-scale version.

The Fruit Stream Cipher: A Tweaked Version of Sprout

- A bit-oriented stream cipher adopting a Grain-like structure and utilizes an 80-bit secret key $K = (k_0, k_1, ..., k_{79})$ and a 70-bit public initial value $IV = (iv_0, iv_1, ..., iv_{69})$.

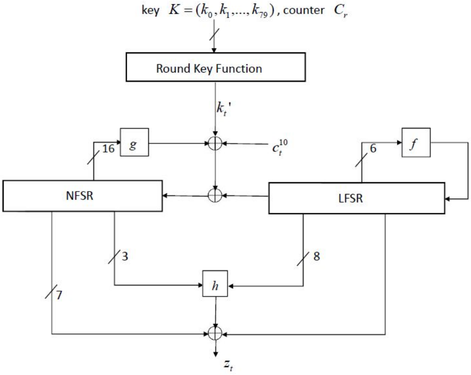

Figure: The keystream generation of Fruit

The Specification (1)

- The 43-bit LFSR is updated independently by a linear function f as $s_{t+43} = f(S^t) = s_t \oplus s_{t+8} \oplus s_{t+18} \oplus s_{t+23} \oplus s_{t+28} \oplus s_{t+37}$.
- The NFSR is updated recursively by a non-linear feedback function g defined as

$$\begin{aligned} n_{t+37} =& k'_t \oplus s_t \oplus c_t^{10} \oplus g(N^t) \\ =& k'_t \oplus s_t \oplus c_t^{10} \oplus n_t \oplus n_{t+10} \oplus n_{t+20} \oplus n_{t+12}n_{t+3} \oplus n_{t+14}n_{t+25} \\ & \oplus n_{t+8}n_{t+18} \oplus n_{t+5}n_{t+23}n_{t+31} \oplus n_{t+28}n_{t+30}n_{t+32}n_{t+34}, \end{aligned}$$

where k'_t is the round key bit, and c_t^{10}, the 4-th LSB of C_c, is the counter bit generated at time t.

The Specification (2)

- Two counter registers, a 7-bit $C_r = (c_t^0, ..., c_t^6)$ and an 8-bit $C_c = (c_t^7, ..., c_t^{14})$, allocated for the round key function and for the initialization/keystream generation, respectively.
- c_t^6 and c_t^{14} are the LSBs of the two counters respectively. These two counters increase by 1 at each tick, and work continually, i.e., after they become all ones, counting from zeros to all ones again.
- Define the values of sv, y, u, p, q, r from the counter C_r as $sv = c_t^0 c_t^1 c_t^2 c_t^3 c_t^4 c_t^5$, $y = c_t^3 c_t^4 c_t^5$, $u = c_t^4 c_t^5 c_t^6$, $p = c_t^0 c_t^1 c_t^2 c_t^3 c_t^4$, $q = c_t^1 c_t^2 c_t^3 c_t^4 c_t^5$ and $r = c_t^3 c_t^4 c_t^5 c_t^6$, then the round key bit k_t' is generated by combining 6 bits of the key as

$$k_t' = k_{sv} k_{y+64} \oplus k_p k_{u+72} \oplus k_{q+32} \oplus k_{r+64}$$

.

The Specification (3)

- Given the internal state (S^t, N^t) at time t, the filter function h is $h_t = n_{t+1}s_{t+15} \oplus s_{t+1}s_{t+22} \oplus n_{t+35}s_{t+27} \oplus n_{t+33}s_{t+11} \oplus s_{t+6}s_{t+33}s_{t+42}$.
- The keystream bit is generated as $z_t = h_t \oplus s_{t+38} \oplus n_t \oplus n_{t+7} \oplus n_{t+13} \oplus n_{t+19} \oplus n_{t+24} \oplus n_{t+29} \oplus n_{t+36}$.
- The details of the initialization phase are omitted here, it is designed in an invertible way to prevent the previous identified weaknesses.

The Generalized Model (1)

- The generalized model is depicted as follows, which is helpful in the sense that we could study some special properties/choices more clearly in a unified framework.

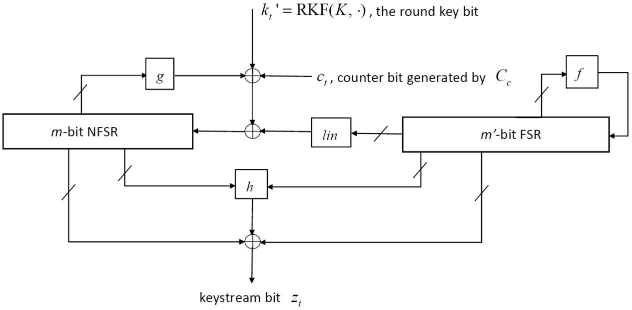

Figure: The generic model for the Grain-like small state stream ciphers

The Generalized model (2)

- $N^t = (n_t, n_{t+1}, ..., n_{t+m-1})$, the m-bit internal state of the cascaded NFSR at time t.
- $S^t = (s_t, s_{t+1}, ..., s_{t+m'-1})$, the m'-bit internal state of the FSR at time t, which updates independently in a invertible way, with a either linear or non-linear feedback function, in the keystream generation phase.
- $K = (k_0, k_1, ..., k_{l-1})$, the l-bit secret key, which satisfies $l \leq m + m' \leq 2l$.
- $k'_t = \mathsf{RKF}(K, \cdot)$, the round key bit generated at time t.
- C_c, a round counter for the NFSR state updating.
- c_t, a counter bit generated by the counter C_c at time t.

The Generalized model (3)

There are five Boolean functions involved in the model

- A (either linear or non-linear) Boolean function f.
- A non-linear Boolean function g.
- A linear Boolean function lin.
- A linear Boolean function ϕ: the linear part of the output function $z_t(\cdot)$.
- A non-linear filter function h, $z_t(\cdot) = h_t(\cdot) \oplus \phi(\cdot)$.
- At each step, the FSR is updated independently by f, while the NFSR is updated by g with the round key bit k'_t, the counter bit c_t, and some bits of the FSR as inputs. The round key bit k'_t at time t is generated by the round key function RKF, which takes the secret key K as part of the input.

The Generalized model (3)

There are five Boolean functions involved in the model

- A (either linear or non-linear) Boolean function f.
- A non-linear Boolean function g.
- A linear Boolean function lin.
- A linear Boolean function ϕ: the linear part of the output function $z_t(\cdot)$.
- A non-linear filter function h, $z_t(\cdot) = h_t(\cdot) \oplus \phi(\cdot)$.
- At each step, the FSR is updated independently by f, while the NFSR is updated by g with the round key bit k'_t, the counter bit c_t, and some bits of the FSR as inputs. The round key bit k'_t at time t is generated by the round key function RKF, which takes the secret key K as part of the input.

The Generalized model (4)

- $P_{S^t} = \{s_{t+\alpha_1}, s_{t+\alpha_2}, ..., s_{t+\alpha_{j_1}}\}$, a subset of S^t and the input variables of the filter function h, from the FSR, $0 \leq \alpha_1 < \alpha_2 < ... < \alpha_{j_1} \leq m' - 1$.
- $P_{N^t} = \{n_{t+\beta_1}, n_{t+\beta_2}, ..., n_{t+\beta_{j_2}}\}$, a subset of N^t and the input variables of the filter function h from the NFSR, $0 \leq \beta_1 < \beta_2 < ... < \beta_{j_2} \leq m - 1$.
- $Q_{S^t} = \{s_{t+\sigma_1}, s_{t+\sigma_2}, ..., s_{t+\sigma_{r_1}}\}$, a subset of S^t and the input variables of the linear Boolean function ϕ, from the FSR, $0 \leq \sigma_1 < \sigma_2 < ... < \sigma_{r_1} \leq m' - 1$.
- $Q_{N^t} = \{n_{t+\eta_1}, n_{t+\eta_2}, ..., n_{t+\eta_{r_2}}\}$, a subset of N^t and the input variables of the linear Boolean function ϕ from the NFSR, $0 \leq \eta_1 < \eta_2 < ... < \eta_{r_2} \leq m - 1..$

The Generalized model (5): Assumed Properties

1. Assume RKF is periodic, so are the round key bits. Let p be the least positive integer such that $k'_{t+p} = k'_t$ for any $t \geq 0$. Besides, our model could also cover the case that the counter bits c_t are unknown. In this case, we only assume that c_t is periodic, i.e., there exists a least positive integer q such that $c_{t+q} = c_t$ for any $t \geq 0$.
2. (Pseudo-linearity) For the filter function $h : \mathrm{GF}(2)^{j_1+j_2} \rightarrow \mathrm{GF}(2)$, $h_{P_{S^t}}(P_{N^t})$ for $P_{S^t} \in \mathrm{GF}(2)^{j_1}$ and $P_{N^t} \in \mathrm{GF}(2)^{j_2}$ is used to replace $h(\cdot)$ for a fixed given value of P_{S^t}. Assume $h_{P_{S^t}}$ to be a *linear* Boolean function with respect to the inputs from P_{N^t}.
3. FSR updates independently, thus for any possible value of the FSR initial state S^0, the outputs of the model depend linearly on the NFSR bits. The degraded system can be interpreted as a linearly filtered NFSR involving the secret round key bits, which have a known cycle p.

Concrete Targets

- The NFSR in the model can be further decomposed into a series of cascaded smaller NFSRs, which could also be treated by our cryptanalysis.
- Grain v1 fits into the model with the parameters $m = 80$, $m' = 80$ and $l = 80$; Fruit fits into the model with the parameters $m = 37$, $m' = 43$ and $l = 80$.
- Plantlet and Lizard do not so far, the reason is that the pseudo -linearity of the corresponding combining functions do not hold in these cases.

Basic Observations and Ideas (1)

- The FSR is updated independently without the influence of the NFSR, the counter bits and the round key bits.
- For small state stream ciphers, the internal state size of the FSR cannot be too large, thus a suitable scale exhaustive search of all the possible values of the independently updated FSR is often feasible.
- Combined with the pseudo-linearity of the h function, we could derive a random probabilistic linear system on the initial NFSR variables with a rather high bias, which will facilitate the construction of low-weight parity-checks to further reduce the dimension of the initial NFSR variables.

Basic Observations and Ideas (2)

- Instead of solving the parity-checks directly: just construct a distinguisher via the well-known FWT and the full Walsh spectrum of some derived function. The FSR is restored independently of the NFSR in the model. This results in a divide-and-conquer recovery of the whole internal state in presence of unknown round key bits.
- The internal state of the NFSR could be retrieved in a multi-pass manner later with a complexity much lower than that of recovering the FSR.
- For the specific ciphers, one period of the round key bits and the original secret key could be derived with a much lower complexity according to the mechanism of the primitive and the definition of the round key function employed.

Outline of Our Attack

Algorithm 1 Fast correlation attack on the generic model

Parameters: m, m', D
Input: A keystream segment $\mathbf{z} = (z_0, z_1, \ldots, z_{D-1})$
1st phase: Prepare the parity-checks

```
for each possible value of LFSR state S^0 do
    use a method to derive the probabilistic system
    construct the parity-checks
end for
```

2nd phase: Recover the full internal state matching with $\mathbf{z}$

```
for each possible value of S^0 do
    use a distinguisher to check it
for each passed candidate of S^0 do
    recover the NFSR state part-by-part
for each candidate of the full internal state do
    check it and restore the secret key accordingly
```

Degrading the System (1)

- If the adversary somehow knows the initial state $S^0 = (s_0, s_1, ..., s_{m'-1})$ of the FSR and the Assumed Properties hold, then he can run the FSR forwards and backwards to remove its protection over the output keystream.
- The resultant system becomes a *linearly* filtered NFSR, involving the periodic round key bits.
- Given the NFSR state $N^t = (n_t, n_{t+1}, ..., n_{t+m-1})$ at time t, we rewrite the keystream bit z_t as

$$z_t = \bigoplus_{i=1}^{j_2} \psi_t^i \cdot n_{t+\beta_i} \oplus \bigoplus_{i=1}^{r_2} n_{t+\eta_i} \oplus \psi_t^0,$$

where the coefficients ψ_t^i, $i = 0, 1, ..., j_2$, depend on the FSR state at time t.

Degrading the System (2)

- For Fruit, the keystream bit generated at time t can be written as

$$\begin{aligned} z_t =& (s_{t+15}\underline{n_{t+1}} \oplus s_{t+11}\underline{n_{t+33}} \oplus s_{t+27}\underline{n_{t+35}}) \\ & \oplus (\underline{n_t} \oplus \underline{n_{t+7}} \oplus \underline{n_{t+13}} \oplus \underline{n_{t+19}} \oplus \underline{n_{t+24}} \oplus \underline{n_{t+29}} \oplus \underline{n_{t+36}}) \\ & \oplus (s_{t+38} \oplus s_{t+1}s_{t+22} \oplus s_{t+6}s_{t+33}s_{t+42}) \end{aligned}$$

  which corresponds to $\psi_t^0 = s_{t+38} \oplus s_{t+1}s_{t+22} \oplus s_{t+6}s_{t+33}s_{t+42}$, $\psi_t^1 = s_{t+15}$, $\psi_t^2 = s_{t+11}$, $\psi_t^3 = s_{t+27}$.

- Even though there is the masking of the secret information, any internal state variable of the NFSR can be expressed as a linear combination of the NFSR state variable at a fixed time instance τ and of some keystream bits, given the FSR initial state S^0.

Degrading the System (3)

For Fruit we have

- $$\begin{aligned} n_{37} = & z_1 \oplus (s_{16}\underline{n_2} \oplus s_{12}\underline{n_{34}} \oplus s_{28}\underline{n_{36}}) \oplus (\underline{n_1} \oplus \underline{n_8} \oplus \underline{n_{14}} \oplus \underline{n_{20}} \oplus \underline{n_{25}} \oplus \underline{n_{30}}) \\ & \oplus (s_{39} \oplus s_2 s_{23} \oplus s_7 s_{34} s_{43}). \end{aligned}$$
- Further, we have

$$\begin{aligned} n_{38} = & (z_2 \oplus s_{29} z_1) \oplus (s_{29} s_{16}\underline{n_2} \oplus s_{17}\underline{n_3} \oplus s_{29} s_{12}\underline{n_{34}} \oplus s_{13}\underline{n_{35}} \oplus s_{29} s_{28}\underline{n_{36}} \\ & \oplus s_{29}\underline{n_1} \oplus \underline{n_2} \oplus s_{29}\underline{n_8} \oplus \underline{n_9} \oplus s_{29}\underline{n_{14}} \oplus \underline{n_{15}} \oplus s_{29}\underline{n_{20}} \oplus \underline{n_{21}} \oplus s_{29}\underline{n_{25}} \\ & \oplus \underline{n_{26}} \oplus s_{29}\underline{n_{30}} \oplus \underline{n_{31}}) \oplus s_{29}(s_{39} \oplus s_2 s_{23} \oplus s_7 s_{34} s_{43}) \\ & \oplus s_{40} \oplus s_3 s_{24} \oplus s_8 s_{35} s_{44}. \end{aligned}$$

1. The effects of the round key bits have been masked successfully.
2. If we carry on this recursive procedure continually, we can get the desirable expressions for n_{37+2}, n_{37+3},...,$n_{37+(D-1)}$ from the keystream bits $z_1, z_2, ..., z_D$, where D is a given parameter.

Building the Parity-checks (1)

Assume there are R linearly independent linear approximations for g having the same largest bias $\epsilon > 0$

- Consider the linear approximation with the sign b_j of the NFSR

$$g(N^t) = \mathbf{a}^j \cdot (N^t)' \oplus b_j = \mathbf{a}^j \cdot (n_t, n_{t+1}, ..., n_{t+m-1})' \oplus b_j,$$

  where $\mathbf{a}^j = (\mathbf{a}_0^j, \mathbf{a}_1^j, \cdots, \mathbf{a}_{m-1}^j)$ for $j = 1, 2, ..., R$ is the linear mask.

- For the inverse process g^{-1} of the NFSR updating function, the corresponding linear approximation is

$$g^{-1}(N^t) = (\mathbf{a}^j \lll 1) \cdot (n_t, n_{t+1}, ..., n_{t+m-1})' \oplus b_j,$$

Building the Parity-checks (2)

- We represent the derived expressions in matrix form as

$$(n_0, n_1, \cdots, n_{m+D-1}) = N^0\mathbf{G} \oplus \chi \oplus \upsilon = (n_0, n_1, ..., n_{m-1})\mathbf{G} \oplus \chi \oplus \upsilon,$$

where the $m \times (m+D)$ matrix $\mathbf{G}$ is formed as $\mathbf{G} = [\mathbf{I}, \mathbf{g}_m, \cdots, \mathbf{g}_{m+D-1}]$ with the first m columns corresponding to the identity matrix $\mathbf{I}$ and $\mathbf{g}_i$ $(m \leq i \leq m+D-1)$ being the column vector, $\chi = (0, 0, \cdots, 0, \chi_m, \cdots \chi_{m+D-1})$, $\upsilon = (0, 0, \cdots, 0, \upsilon_m, \cdots, \upsilon_{m+D-1})$ are $(m+D)$-bit vectors depending on the FSR initial state and the keystream bits $z_{m-\eta_{r_2}+i}$ for $0 \leq i \leq D-1$.

- Then for $j = m, ..., m+D-1$, we have

$$n_j = N^0 \cdot \mathbf{g}_j \oplus \chi_j \oplus \upsilon_j = (n_0, n_1, ..., n_{m-1}) \cdot \mathbf{g}_j \oplus \chi_j \oplus \upsilon_j,$$

where χ_j and υ_j are the jth coordinates of χ and υ, respectively.

Building the Parity-checks (3)

- For Fruit, the counter bit c_t^{10} is known and has the period $q = 32$, while the round key bit k'_t has the period $p = 128$.
- By looking at the equations at an interval of 128, we could derive the following equations.
- For each possible LFSR state, we can obtain a linear system with $\omega' = 7 \cdot \omega$ linear equations, all holding with the bias $\epsilon = 2^{-4.6}$ and $b_j = 0$ for $1 \leq j \leq 7$ and $i = 0, 1, ..., \omega - 1$,

$$(n_0, n_1, \cdots, n_{36}) \cdot \mathbf{u}_{i,j} \oplus Z_{i,j} \oplus \mathbf{v}_{i,j} = k'_0 \oplus c_0^{10} \oplus e_{i,j},\ j = 1, 2, ..., 7,$$

Here $Z_{i,j}$ depends on the keystream and column vecors $\mathbf{u}_{i,j}$ are determined by the FSR initial state.

Constructing the Parity-checks

- For Fruit, we look for some κ-tuple of (usually $\kappa = 2$ or $\kappa = 4$ to cancel the secret information) column vectors $(\mathbf{u}_{i_1,j_1}, ..., \mathbf{u}_{i_\kappa,j_\kappa})$ satisfying $Low_{m-m_1}(\mathbf{u}_{i_1,j_1} \oplus ... \oplus \mathbf{u}_{i_\kappa,j_\kappa}) = (0, ..., 0)'$.
- Denote the t-th pair of columns by $(\mathbf{u}^{(t)}_{i_1,j_1}, \mathbf{u}^{(t)}_{i_2,j_2})$ for $t = 1, 2, ..., \Omega$. Similarly we define the notations that $\mathcal{Z}_t = \mathbf{Z}^{(t)}_{i_1,j_1} \oplus \mathbf{Z}^{(t)}_{i_2,j_2}$, $\mathcal{V}_t = \mathbf{v}^{(t)}_{i_1,j_1} \oplus \mathbf{v}^{(t)}_{i_2,j_2}$, $\mathcal{E}_t = e^{(t)}_{i_1,j_1} \oplus e^{(t)}_{i_2,j_2}$ and $\mathcal{U}_t = High_{m_1}\left(\mathbf{u}^{(t)}_{i_1,j_1} \oplus \mathbf{u}^{(t)}_{i_2,j_2}\right)$, thus we derive $\Omega = \omega'^2 \cdot 2^{-(m-m_1+1)}$ equations as follows,

$$(n_0, n_1, ..., n_{m_1-1}) \cdot \mathcal{U}_t \oplus \mathcal{Z}_t \oplus \mathcal{V}_t = \mathcal{E}_t,\ t = 1, 2, ..., \Omega$$

Here $\Pr(\mathcal{E}_t = 0) = \frac{1}{2} + 2\epsilon^2 \triangleq \frac{1}{2}(1 + \epsilon_F)$, where $\epsilon = 2^{-4.6}$ and $\epsilon_F = 4\epsilon^2 = 2^{-7.2}$ for $\kappa = 2$.

The Attack Profile

- Make an independent guess/recovery of the FSR initial state S^0.
- Restore the NFSR state with the multi-pass strategy, given the candidates of the FSR state.
- Recover the secret information bits within one cycle.

Restoring the Internal State of the FSR (1)

- Denote by α the probability that the correct guess $\mathbf{s}_c$ will be chosen as a candidate, and by β the probability that a wrong guess $\mathbf{s}_w$ would be chosen as a candidate, then

$$\alpha = \Pr(\mathcal{F}(\mathbf{s}_c) \geq T) = 1 - \Phi\left(\frac{T - \Omega\epsilon_F}{\sqrt{\Omega(1-\epsilon_F^2)}}\right),$$

$$\beta = \Pr(\mathcal{F}(\mathbf{s}_w) \geq T) = 1 - \Phi\left(\frac{T}{\sqrt{\Omega}}\right) \triangleq 2^{a'}.$$

In cryptanalysis, we expect to choose a T such that α is very close to 1 to assure a high passing probability for the correct guess, meanwhile β is very small to filter out all the wrong guesses, or to reduce the passing number of wrong guesses as much as possible.

Restoring the Internal State of the FSR (2)

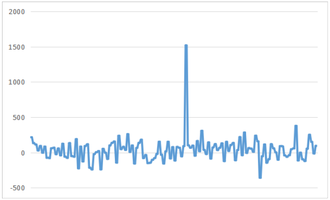

Figure: Walsh Spectrum of derived function $h_{\mathbf{s}_c}$ for the correct guess of S^0

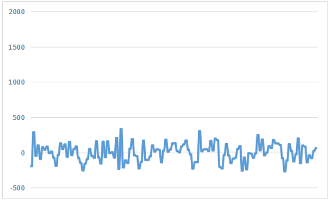

Figure: Walsh Spectrum of derived function $h_{\mathbf{s}_w}$ for a random wrong guess of S^0

Restoring the Secret Information Bits within One Cycle (1)

Algorithm 2

Input: a state candidate (S^0, N^0).

Output: a flag representing the correctness of the state candidate, and output $k'_i \oplus c_i,\ i = 0, 1, ..., d-1$, for the correct one.

```
Create a d-bit vector ζ;
for i = 0, 1, ..., d − 1 do
    compute n_{m+i} from z_{m−η_{r_2}}, z_{m−η_{r_2}+1}, ..., z_{m−η_{r_2}+i};
    compute k'_i ⊕ c_i = n_{m+i} ⊕ lin(S^i) ⊕ g(n_i, n_{1+i}, ..., n_{m−1+i});
    store k'_i ⊕ c_i at the i-th position of the vector ζ, i.e., ζ[i] = k'_i ⊕ c_i.
for i = 0, 1, ..., d − 1 do
    compute n_{m+d+i} from z_{m+d−η_{r_2}}, z_{m+d−η_{r_2}+1}, ..., z_{m−η_{r_2}+d+i};
    compute v_i ≜ n_{m+d+i} ⊕ lin(S^{d+i}) ⊕ g(n_{d+i}, n_{1+d+i}, ..., n_{m−1+d+i});
    if v_i = ζ[i] then continue for next i;
    else output a flag that the state candidate is wrong and stop.
if v_i = ζ[i] for all i = 0, 1, ..., d − 1
  then output a flag that the state candidate is correct,
       and output the d secret information bits, i.e., ζ[i], i = 0, 1, ..., d − 1.
```

Restoring the Secret Information Bits within One Cycle (2)

The Fruit case

- For any state candidate, the average number of ticks for state checking is $d + (1 \cdot \frac{1}{2^0} + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{2^2} + ... + d \cdot \frac{1}{2^{d-1}}) \approx d + 4$.
- For Fruit, we plug in the corresponding parameters into Alg.2, and obtain an algorithm for recovering the 128 round key bits, by combining the fact that the counter bits c_t^{10} are known at any time t. The average number of ticks for state checking is 132.

Complexity Analysis (1)

The Fruit case

1. Set $m_1 = 21$, i.e., we divide the NFSR into two parts of length 21 bits and $37 - 21 = 16$ bits, respectively.
2. Let $\omega = 2^{16.35}$, number of linear equations, and data $D = 128(\omega - 1) + 1 = 2^{23.35}$.
3. By using the 7 best linear approximations for g, we can construct $\omega' = 7 \cdot \omega = 2^{19.16}$ parity checks containing the full NFSR initial state variables, from which we can construct another $2^{21.32}$ parity checks containing only the first 21 variables of the NFSR initial state.

With suitable parameters, the time complexity for recovering the 80-bit secret key of Fruit is $2^{70.55}$, equivalent to $2^{62.81}$ Fruit encryptions.

Complexity Analysis (2): Two New Design Criteria

Based on the theoretical framework established, we have the following design criteria on Grain-like small state stream ciphers.

1. The pseudo-linearity of the output function when combining the input variables should be avoided.
2. For l-bit security, there should exist no linear approximation with the bias ϵ for the state updating function g of the NFSR such that the resulting $D < 2^l$ and $C < 2^l$, where C is the time complexity and D is the data complexity.

Experimental results

A reduced version of Fruit:

- A 19-bit LFSR whose state at time t is denoted by $S^t = (s_t, s_{t+1}, ..., s_{t+18})$, a linked 18-bit NFSR whose state at time t is denoted by $N^t = (n_t, n_{t+1}, ..., n_{t+17})$, a 37-bit fixed key register, and two counter registers: a 6-bit counter $C_r = (c_t^0, ..., c_t^5)$ and a 7-bit counter $C_c = (c_t^6, ..., c_t^{12})$.
- The 19-bit LFSR is updated independently and recursively as $s_{t+19} = s_t \oplus s_{t+3} \oplus s_{t+7} \oplus s_{t+17}$.
- The 18-bit NFSR is updated recursively by a non-linear feedback function g defined as $n_{t+18} = k'_t \oplus s_t \oplus c_t^9 \oplus g(N^t)$, where $g(N^t) = n_t \oplus n_{t+5} \oplus n_{t+10} \oplus n_{t+12}n_{t+3} \oplus n_{t+2}n_{t+13}n_{t+15}$, and c_t^9 is the 3-th LSB of the counter C_c.

The experiments match the theoretical results quite well in the simulations.

Conclusions

- We have studied the security of Grain-like small state stream ciphers by fast correlation attacks, the classical cryptanalytic method against LFSR-based stream ciphers. $\rightarrow$ traditional methods still work
- A formal framework for fast correlation attacks utilizing the divide-and-conquer strategy on the generic model is presented with a thorough theoretical analysis.
- If the non-linear combining function has some pseudo-linear property when combining the input variables from the cascaded internal state, then such an attack would be applicable in principle. $\rightarrow$ new general design criteria
- We break Fruit, a tweaked version of Sprout, in $2^{62.8}$ Fruit encryptions, given $2^{22.3}$ keystream bits for all the keys, which clearly violates the 80-bit security claim. Our results have been verified in experiments on a small-scale version of Fruit.

Thank you!

Q & A