Fast Correlation Attack Revisited

Cryptanalysis on Full Grain-128a, Grain-128, and Grain-v1

Yosuke Todo[1], Takanori Isobe[2], Willi Meier[3], Kazumaro Aoki[1], Bin Zhang[4]

1: NTT Secure Platform Laboratories

2: University of Hyogo

3: FHNW

4: Chinese Academy of Science

Fast correlation attack

- One of the most traditional attacks.
  - The initial idea was proposed in 80's.
    - Correlation attack [Siegenthaler, 1985]
    - Fast correlation attack [Meier and Staffelbach,1989]
- We revisit the fast correlation attack.
  - New property, wrong-key hypothesis, and attack framework
  - Attack against full **Grain-v1** (eSTREAM) and **Grain-128a** (ISO/IEC 29167-13).
    - Grain-v1 : with about $2^{76.4}$ time complexity.
    - Grain-128a : with about 2^{115} time complexity.

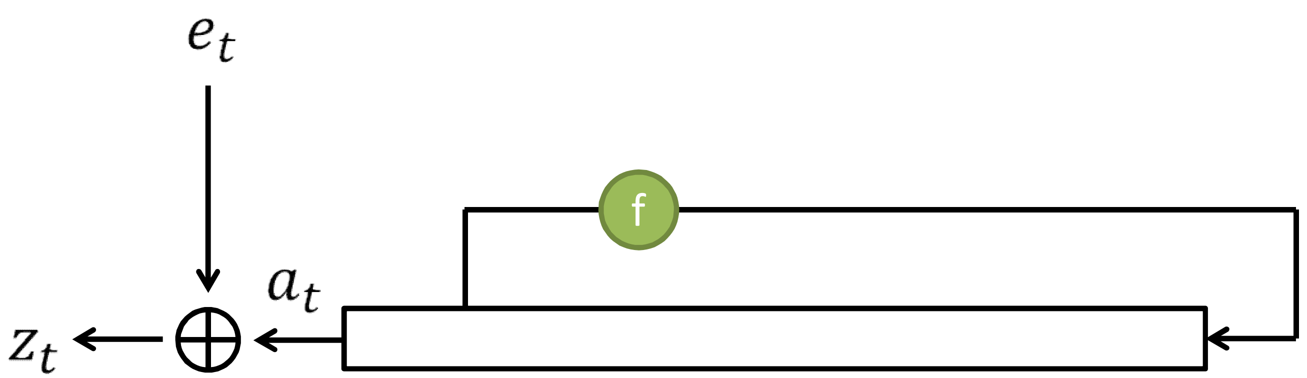
Innovative R&D by NTT

NTT

Innovative R&D by NTT

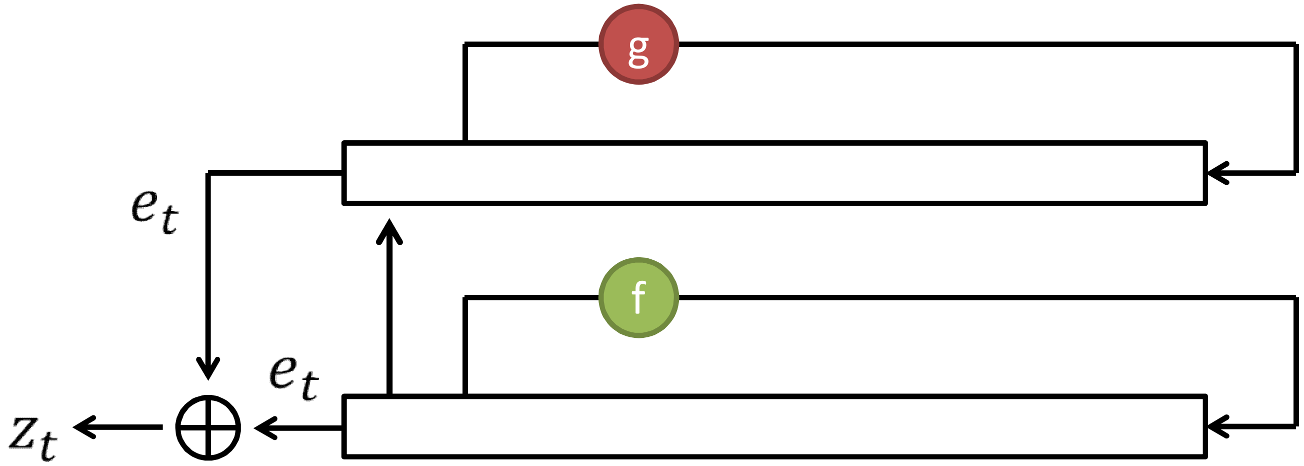
Preliminaries

LFSR-based stream ciphers



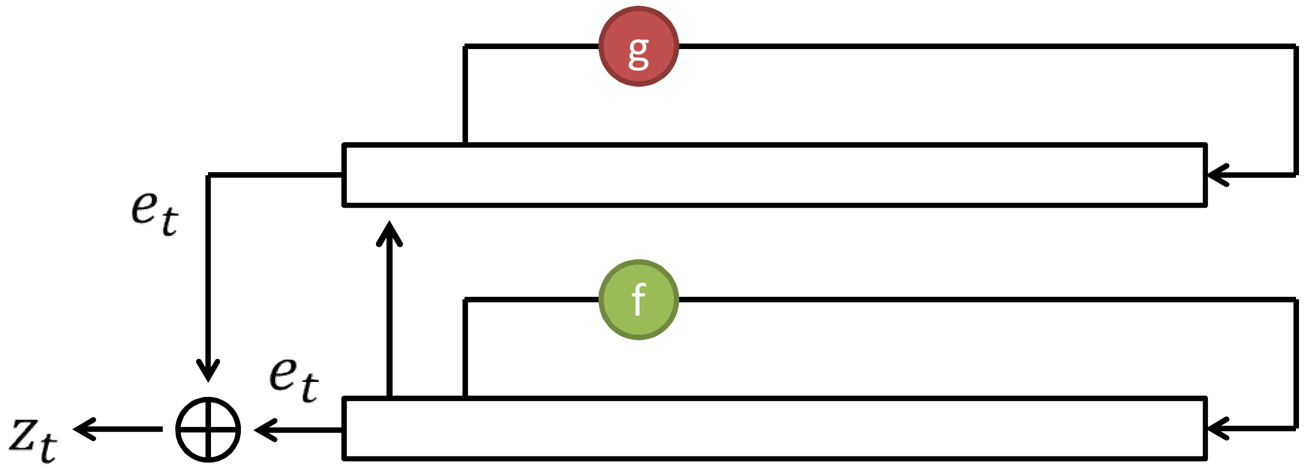
- The key stream sequence is generated by the XOR between the output sequence of the LFSR and error.
  - Error is very important because the internal state can be recovered efficiently if $e_t = 0$ for all time.

More practical LFSR-based stream ciphers.



- The error sequence is **nonlinearly** generated from another internal state.

Correlation attack



- Assume $(\Pr[e_t{=}0] - \Pr[e_t{=}1]) = c$.
- Guess initial state st_0 and compute $a_t = \langle st_0, \Lambda_t \rangle$.
- If we guess correct st_0, $a_t \oplus z_t$ coincides with e_t.

$$\sum_{t=0}^{N-1} (-1)^{a_t \oplus z_t} \sim \begin{cases} \mathcal{N}(Nc, N) & \text{for correct guess.} \\ \mathcal{N}(0, N) & \text{for incorrect guess.} \end{cases}$$

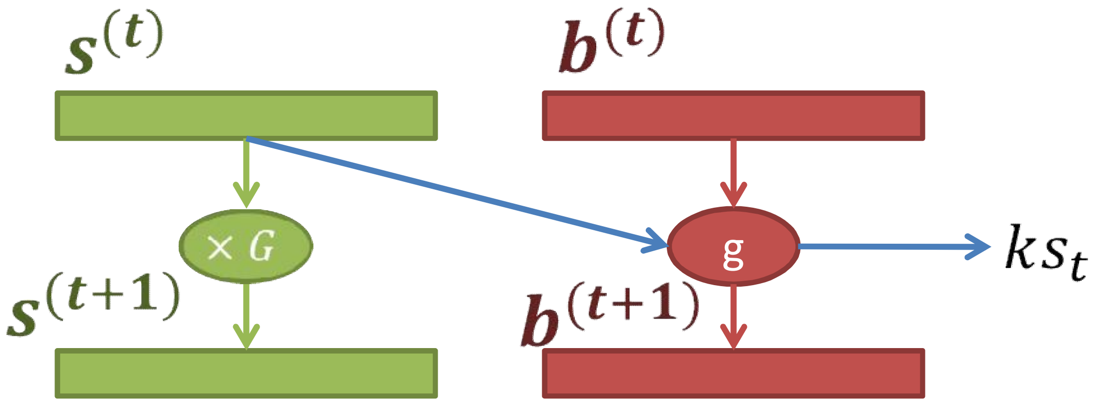
Correlation attack

- Usually, e_t is not biased in the modern stream cipher.
- However, $\bigoplus_{i \in \mathbb{T}_z} z_{t+i}$ may have biased relation with the initial state by optimally choosing $\mathbb{T}$.
  - Known results.
    - Grain v0 [Berbain et al, FSE2006]
    - Sosemanuk and SNOW2.0 [Lee et al, AC08]
    - SNOW2.0 [Zhang et al, CRYPTO15]
  - For example, Berbain et al uses $\mathbb{T}_z = \{0,80\}$ to attack Grain v0.

Procedure of FCA

1. Generating parity check equations.
2. Reduce the size of secret bits involved to parity check equations.
3. Recover involved secret bits by using parity check equations.
   - FWHT is applied to accelerate this part.

1. Generating parity check equations from linear trail

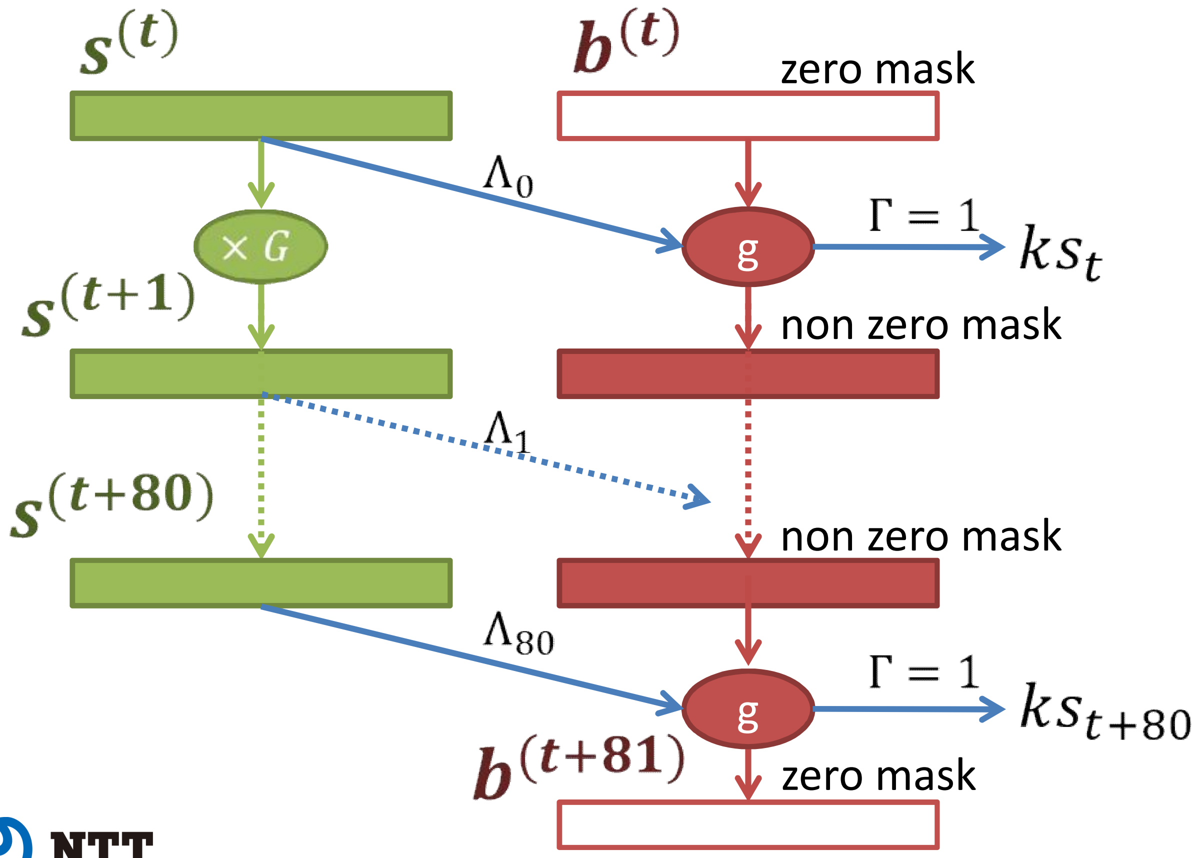


1. Generating parity check equations from linear trail

e.g., Case of $\mathbb{T}_z = \{0,80\}$

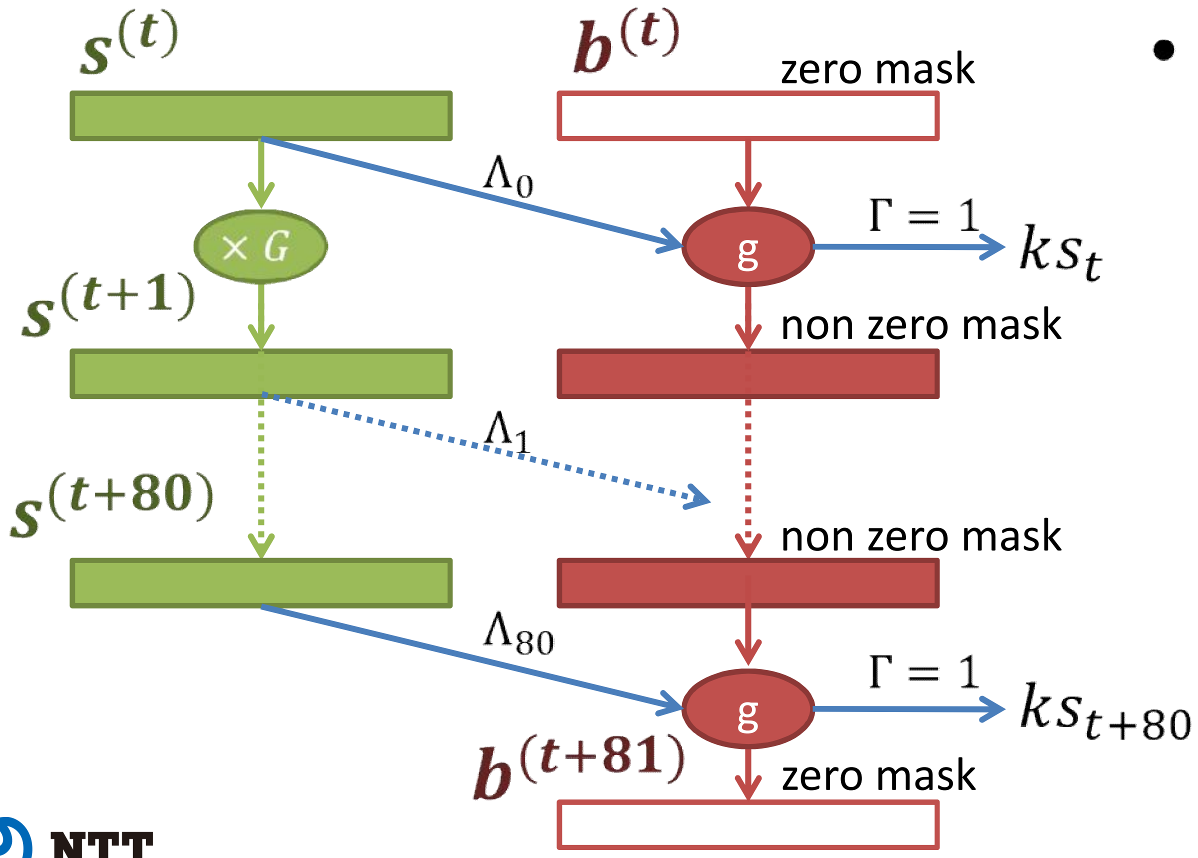
- High-biased linear trail.

$$\bigoplus_{i\in\mathbb{T}_z} \langle s^{(t+i)}, \Lambda_i \rangle \approx \bigoplus_{i\in\mathbb{T}_z} z_{t+i}$$



1. Generating parity check equations from linear trail

e.g., Case of $\mathbb{T}_z = \{0,80\}$



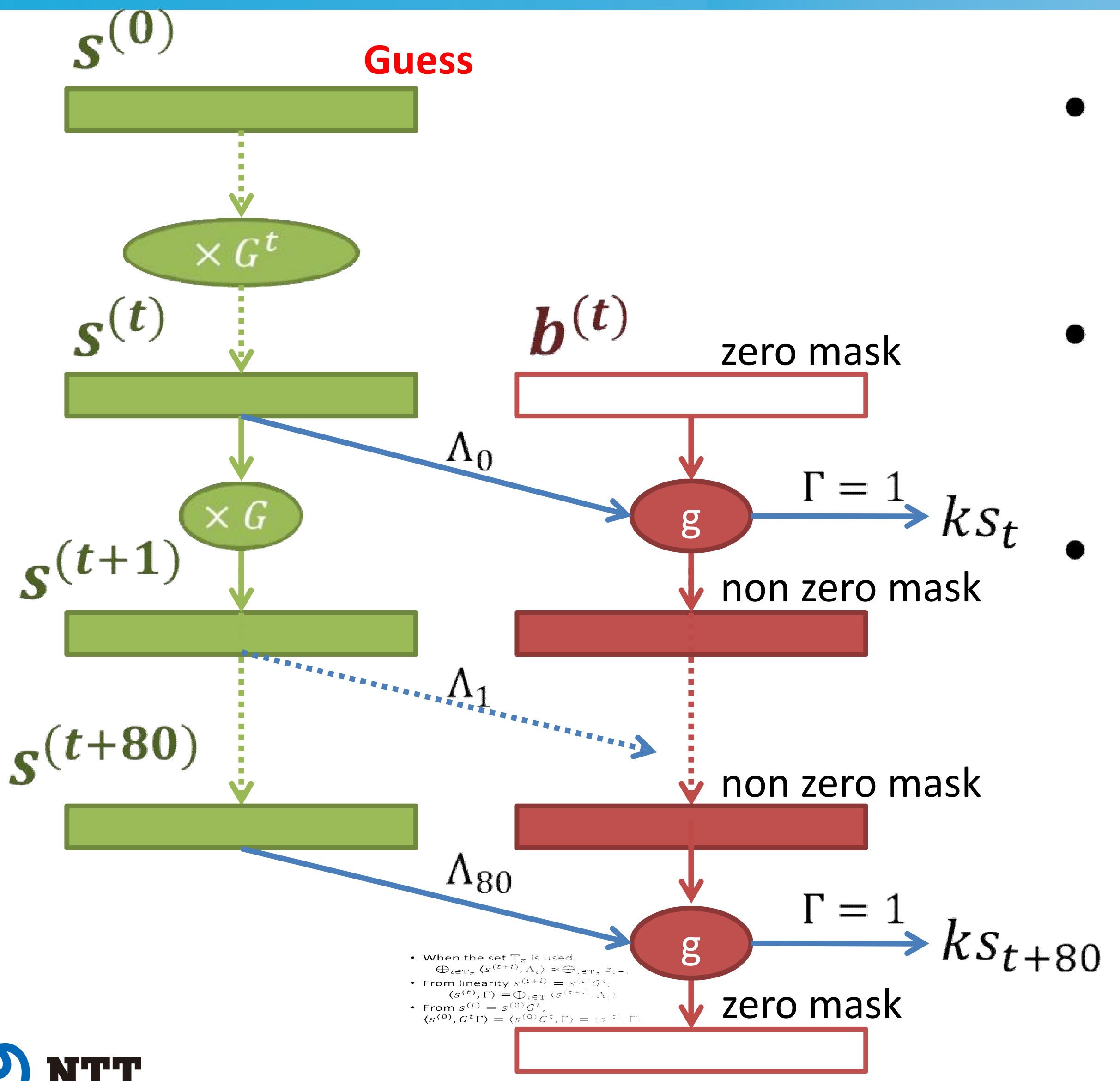
- High-biased linear trail.

$$\bigoplus_{i \in \mathbb{T}_z} \langle s^{(t+i)}, \Lambda_i \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

- From linearity $s^{(t+i)} = s^{(t)} G^i$,

$$\langle s^{(t)}, \Gamma \rangle = \bigoplus_{i \in \mathbb{T}} \langle s^{(t+i)}, \Lambda_i \rangle$$

1. Generating parity check equations from linear trail



- When the set $\mathbb{T}_z$ is used,

  $$\bigoplus_{i \in \mathbb{T}_z} \langle s^{(t+i)}, \Lambda_i \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

- From linearity $s^{(t+i)} = s^{(t)} G^i$,

  $$\langle s^{(t)}, \Gamma \rangle = \bigoplus_{i \in \mathbb{T}} \langle s^{(t+i)}, \Lambda_i \rangle$$

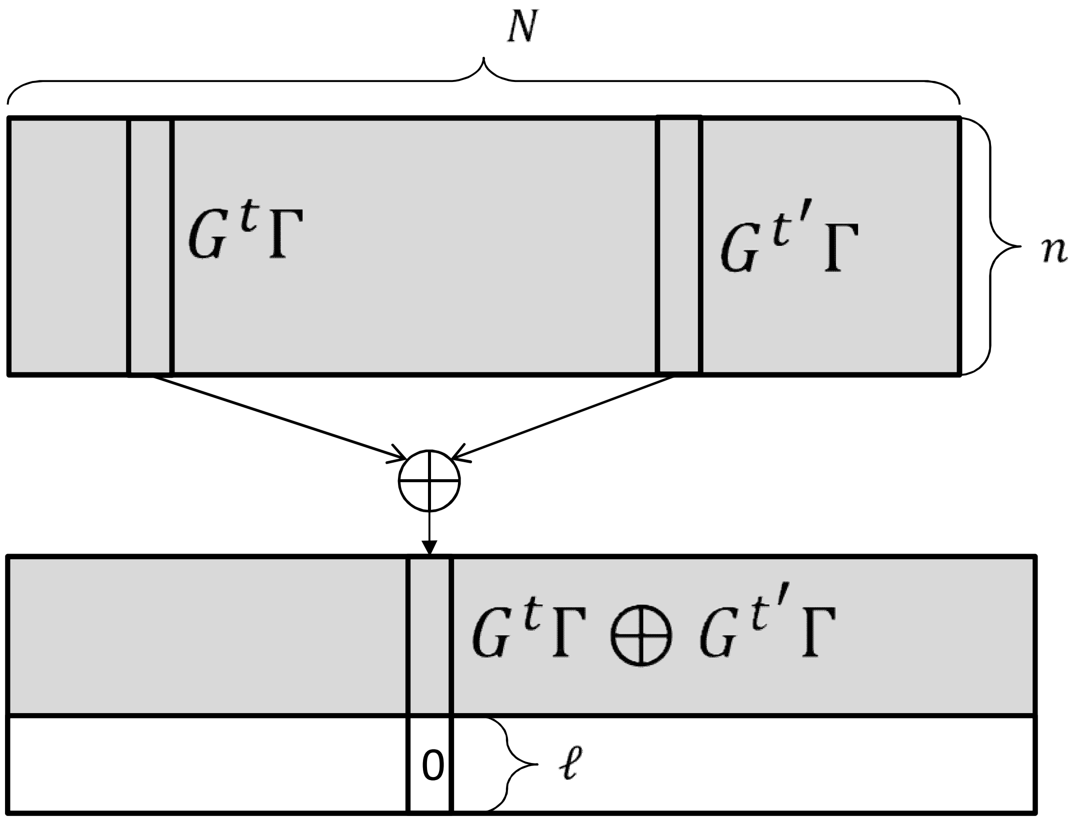
- From $s^{(t)} = s^{(0)} G^t$,

  $$\langle s^{(t)}, \Gamma \rangle = \langle s^{(0)}, G^t \Gamma \rangle$$

Linear approximations

$$\langle s^{(0)}, G^t \Gamma \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

2. Reduce involved secret-key bits



Original linear approximations

$$\langle s^{(0)}, G^t\Gamma\rangle \approx \bigoplus_{i\in\mathbb{T}_z} z_{t+i}$$

Solve partial birthday problem.

New linear approximations

$$\langle s^{(0)}, G^t\Gamma \oplus G^{t'}\Gamma\rangle \approx \bigoplus_{i\in\mathbb{T}_z} (z_{t+i} \oplus z_{t'+i})$$

We don't need to guess last ℓ bits of $s^{(0)}$.

3. Recover $s^{(0)}$

- Recover $s^{(0)}$ such that

$$\sum_{(t,t')\in S} \left(\langle s^{(0)}, G^t\Gamma \oplus G^{t'}\Gamma \rangle + \bigoplus_{i\in\mathbb{T}_z} (z_{t+i} \oplus z_{t'+i}) \right)$$

is farthest from $|S|/2$.

  - The trivial procedure requires $|S|2^{n-\ell}$.
  - FWHT can evaluate it with $|S| + (n-\ell)2^{n-\ell}$.

- After recovery of partial $s^{(0)}$,
  - Recover full $s^{(0)}$, and then $b^{(0)}$.
  - The secret key is recovered from inverse key initialization

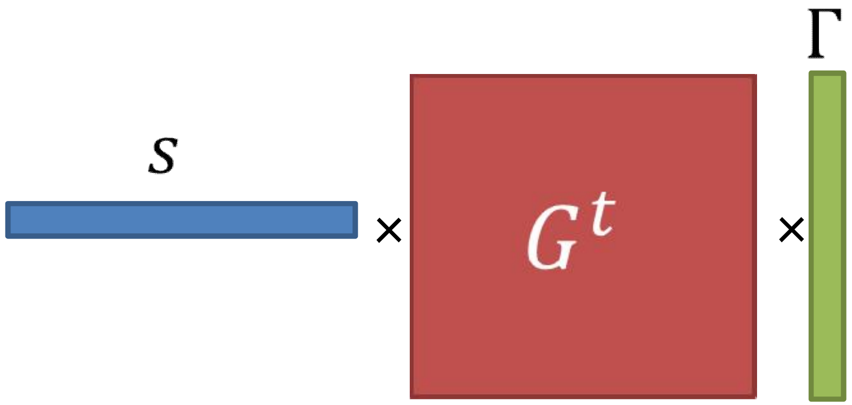
Drawback

- The correlation drops because of the birthday problem.
  - Let c be the original correlation.
  - The correlation of the new one is c^2.

- The rough estimation of required data is $O(1/c^4)$.
  - Even if we find linear approximation with correlation 2^{-50}, the required data is about 2^{200}.
  - In the case of Grain-128a, the correlation must be larger than 2^{-32}.

New Insight of Fast Correlation Attack

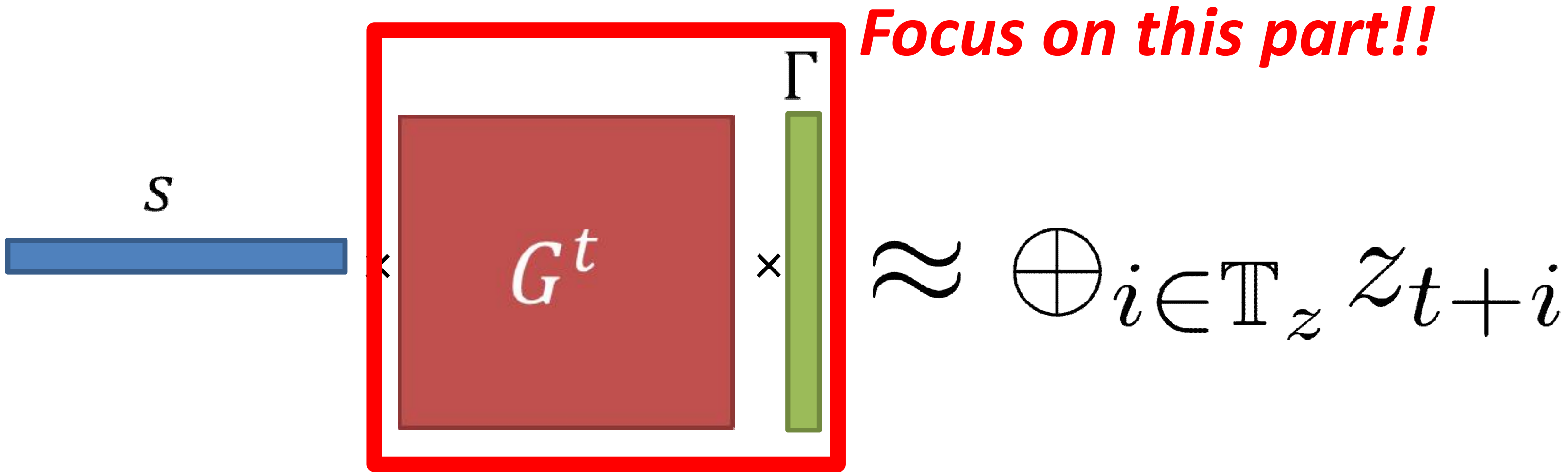
First step



$$s \times G^t \times \Gamma \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

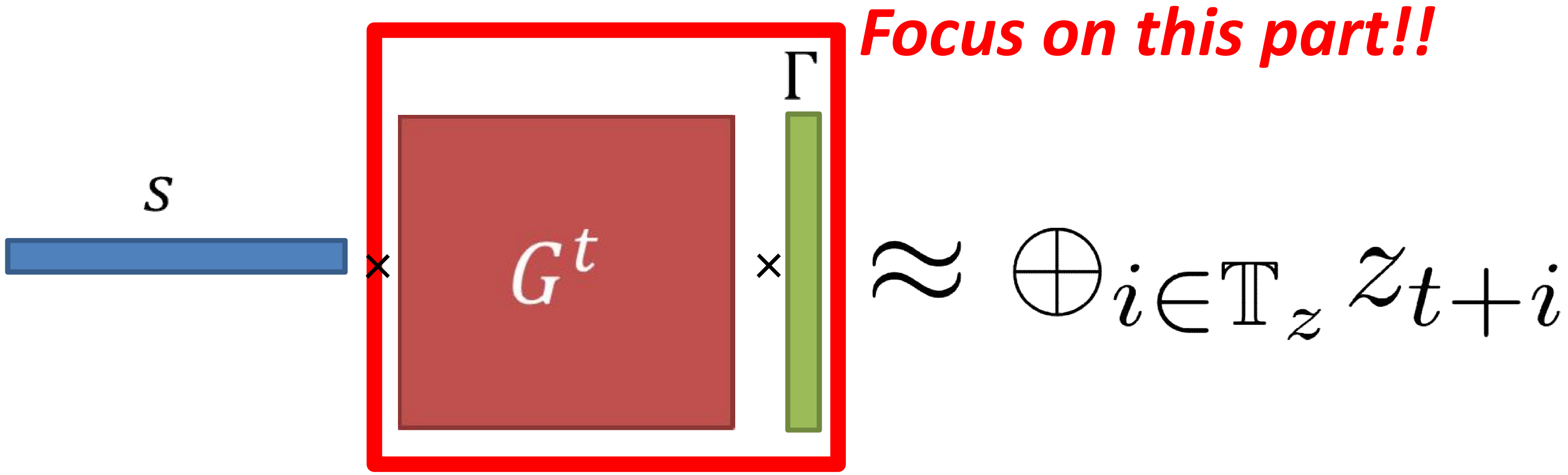
- s is an n-bit row vector and secret.
  - For simplicity, we rewrite $s^{(0)}$ as s.
- G is the $n \times n$ matrix representation of LFSR.
- Γ is an n-bit linear mask.

Focus on $G^t \times \Gamma$



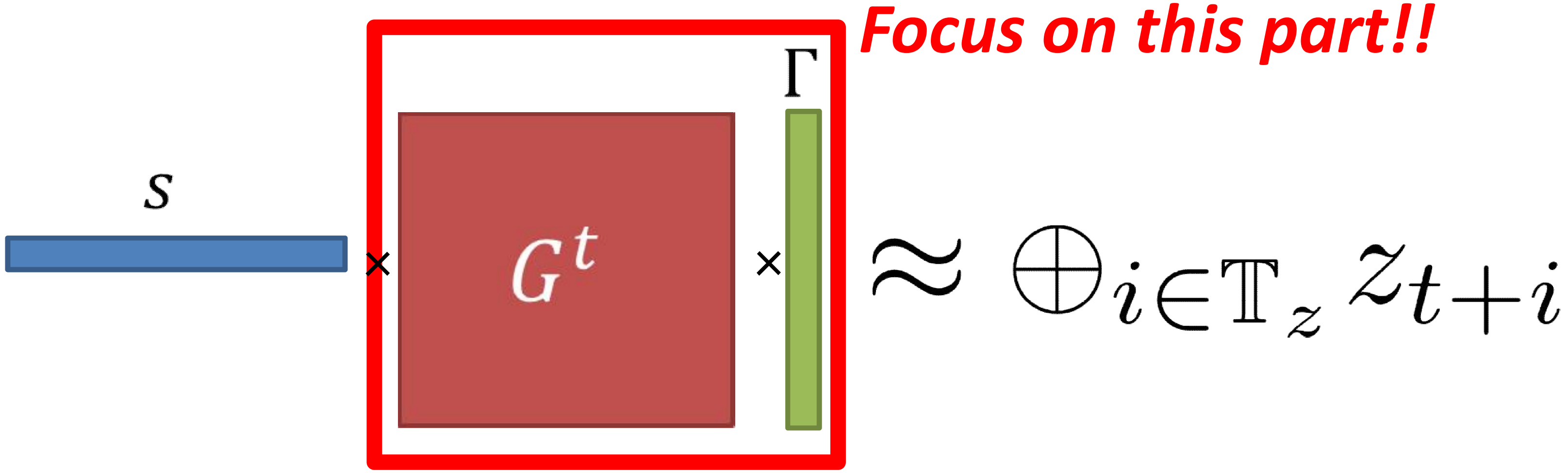
- s is an n-bit row vector and secret.
  - For simplicity, we rewrite $s^{(0)}$ as s.
- G is the $n \times n$ matrix representation of LFSR.
- Γ is an n-bit linear mask.

Link to $GF(2^n)$



- Let's consider the finite field $GF(2^n)$.
  - The primitive polynomial is the feedback polynomial of LFSR.
  - Let $\alpha \in GF(2^n)$ be the primitive root.
  - $\gamma \in GF(2^n)$ is natural conversion from $\Gamma \in \{0,1\}^n$.

$G^t \times \Gamma$ is "commutative"



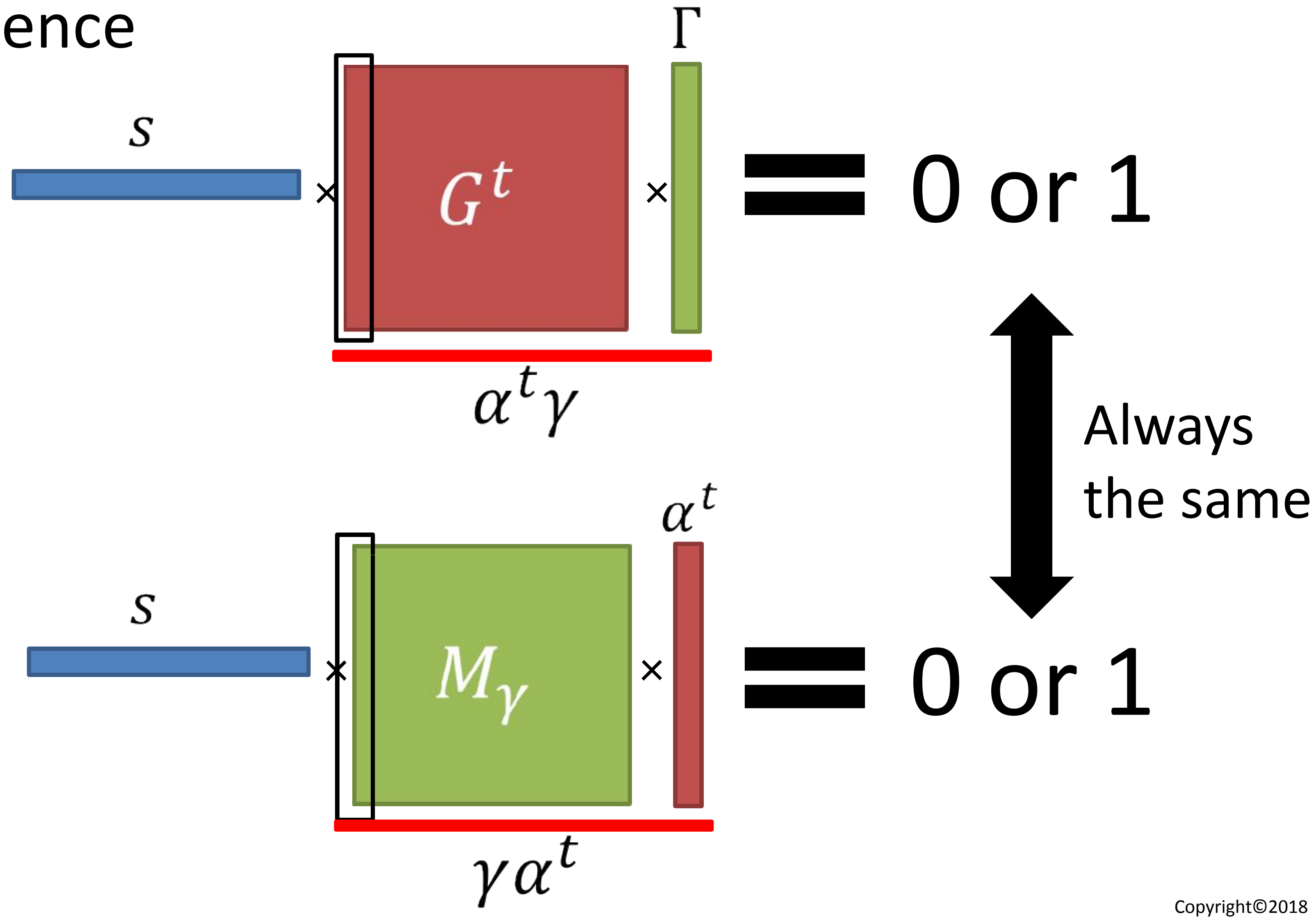
- $G^t \times \Gamma \in \{0,1\}^n$ is natural conversion of $\alpha^t \gamma \in GF(2^n)$.

- Multiplication over $GF(2^n)$ is commutative.

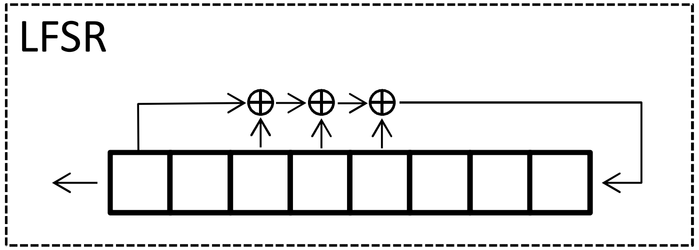
$$G^t \times \Gamma \;\Leftrightarrow\; \alpha^t \gamma \Leftrightarrow \gamma \alpha^t \Leftrightarrow M_\gamma \times \alpha^t$$

$G^t \times \Gamma$ is "commutative"

- Equivalence



Structure of M_γ



$$\gamma = \alpha + \alpha^3 + \alpha^4 + \alpha^6 + \alpha^7$$
$$(\Gamma = 01011011)$$

Corresponding Galois Field

$$GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x^2 + 1)$$

γ $\gamma\alpha$ $\gamma\alpha^7$

$$M_\gamma = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

New property for FCA

$$s \times G^t \times \Gamma = \langle s, G^t \times \Gamma \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

$$\langle s \times M_\gamma, \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

where α^t is converted into an element over $\{0,1\}^n$ naturally.

New property for FCA

$$\langle s \times M_\gamma, \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$$

where α^t is converted into an element over $\{0,1\}^n$ naturally.

- Parity check equations are generated from α^t.
  - If attackers guess $s \times M_\gamma$ instead of s, the approximation above holds with high probability.
  - If there are m high-biased masks $\gamma_1, \dots, \gamma_m$, all of $s \times M_{\gamma_i}$ are highly biased.

We have multiple biased solutions!!

NTT

New Algorithm for the FCA

Motivation

- We want to exploit multiple solutions.

Conventional FCA $\langle s', G^t \times \Gamma \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$

If $s' = s$, the approximation above holds w.h.p.

New FCA $\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}_z} z_{t+i}$

If $s' = s \times M_\gamma$, the approximation above holds w.h.p.
Multiple γ implies multiple high-biased solutions.

Basic framework of new algorithm

1. Generate parity-check equations.
   $\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}} z_{t+i}$ for $\mathrm{t} \in \{0, 1, \ldots, \mathrm{N}-1\}$

2. Pick top N_p s' whose empirical bias is high.
   $s' = s \times M_{\gamma_i}$

3. Recover s from picked s'.
   $s' \times M_{\gamma_i}^{-1} = s$

Basic framework of new algorithm

1. Generate parity-check equations.
   $\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}} z_{t+i}$ for t $\in \{0, 1, \ldots, \mathrm{N}-1\}$
   Time complexity : *N*

2. Pick top N_p s' whose empirical bias is high.
   $s' = s \times M_{\gamma_i}$

3. Recover s from picked s'.
   $s' \times M_{\gamma_i}^{-1} = s$

NTT

Basic framework of new algorithm

1. Generate parity-check equations.

   $$\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}} z_{t+i} \text{ for t} \in \{0, 1, \ldots, \mathrm{N}-1\}$$

   Time complexity : N

2. Pick top N_p s' whose empirical bias is high.

   $$s' = s \times M_{\gamma_i}$$

   Time complexity : $n2^n$

   Exceeds to 2^n

3. Recover s from picked s'.

   $$s' \times M_{\gamma_i}^{-1} = s$$

Bypassed FWHT

- We don't need to evaluate all $s' \in GF(2)^n$.
  - Because there are multiple s's with high bias.
  - Even if we only evaluate s' whose LSB is always 0, we should find $m/2$ high-biased s's on average.
  - Complexity of the bypassed FWHT is $(n-1)2^{n-1}$.
  - If β bits are bypassed, it reduces to $(n-\beta)2^{n-\beta}$.

Basic framework of new algorithm

1. Generate parity-check equations.
   $\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}} z_{t+i}$ for $t \in \{0, 1, \ldots, N-1\}$
   Time complexity : N

2. Pick top N_p s' whose empirical bias is high.
   $s' = s \times M_{\gamma_i}$
   Time complexity : $(n-\beta)2^{n-\beta}$
   β-bit bypass

3. Recover s from picked s'.
   $s' \times M_{\gamma_i}^{-1} = s$

Basic framework of new algorithm

1. Generate parity-check equations.
   $\langle s', \alpha^t \rangle \approx \bigoplus_{i \in \mathbb{T}} z_{t+i}$ for $t \in \{0, 1, \ldots, N-1\}$
   Time complexity : N
2. Pick top N_p s' whose empirical bias is high.
   $s' = s \times M_{\gamma_i}$
   Time complexity : $(n - \beta)2^{n-\beta}$
   β-bit bypass
3. Recover s from picked s'.
   $s' \times M_{\gamma_i}^{-1} = s$
   Time complexity : $N_p \times m$
   m: # of biased masks

NTT

How to estimate the success probability

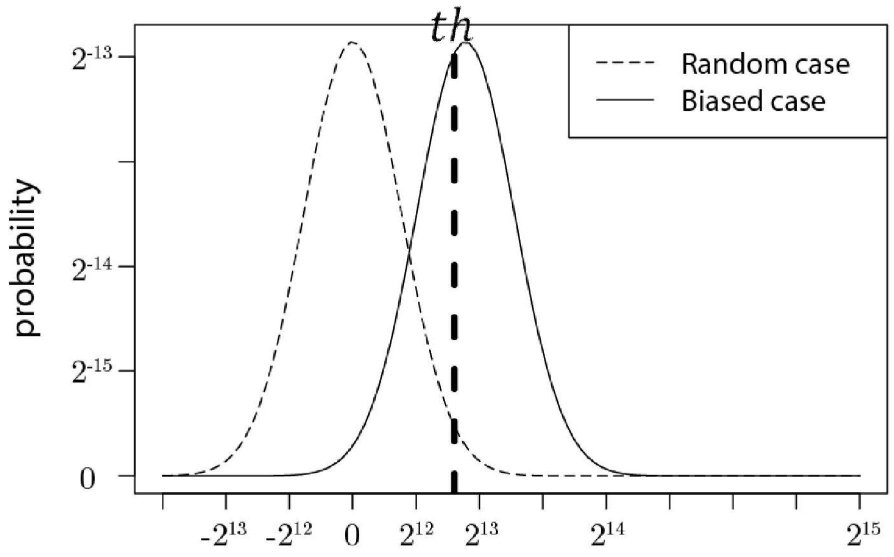
1. Generate parity-check equations.

2. Pick top N_p s' whose empirical bias is high.
   - Normal distributions are used.

3. Recover s from picked s'.
   - Poisson distributions are used.

Example parameter

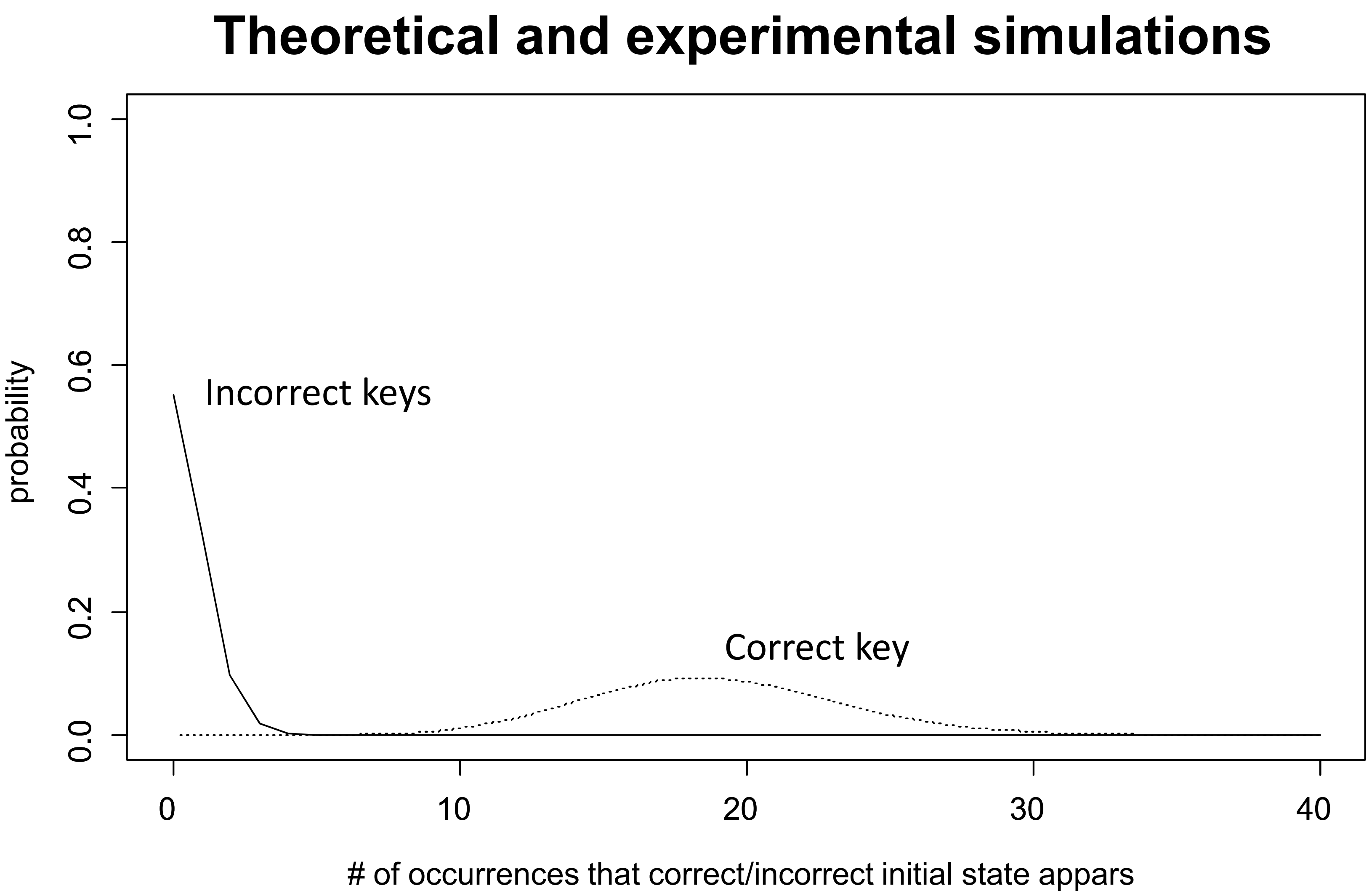
- $(n, c, m, \beta, th) = (24, 2^{-10.415}, 2^{10}, 5, 2^{12.68})$
- Collect $N = 2^{23.25}$, and evaluate $\sum_{t=0}^{N} (-1)^{\langle s', \alpha^t \rangle \oplus_{i \in \mathbb{T}_z} z_{t+i}}$
- Normal distributions



- If $s' = s \times M_{\gamma_i}$, $\mathcal{N}(Nc, N)$.
  - Let ϵ_b be the probability s.t. it's greater than th.
- Otherwise, $\mathcal{N}(0, N)$.
  - Let ϵ_u be the probability s.t. it's greater than th.

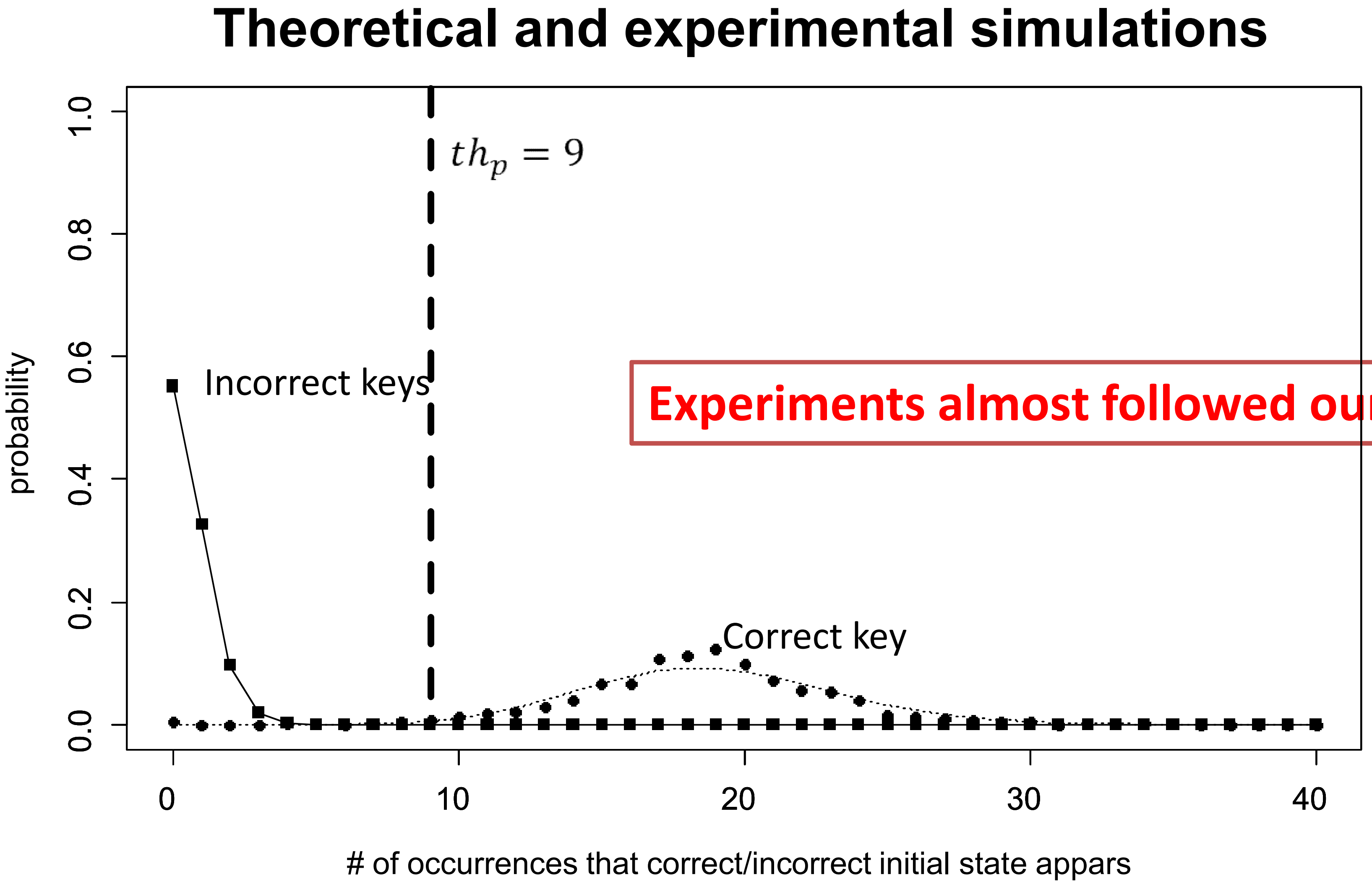
$N_p = m2^{-\beta}\epsilon_b + 2^{n-\beta}\epsilon_u \approx 2^{n-\beta}\epsilon_u = 2^{13.28}$ solutions are left.

Theoretical estimation (Poisson distribution)



Plot experimental results (Poisson distribution)

Average of 1000 trails

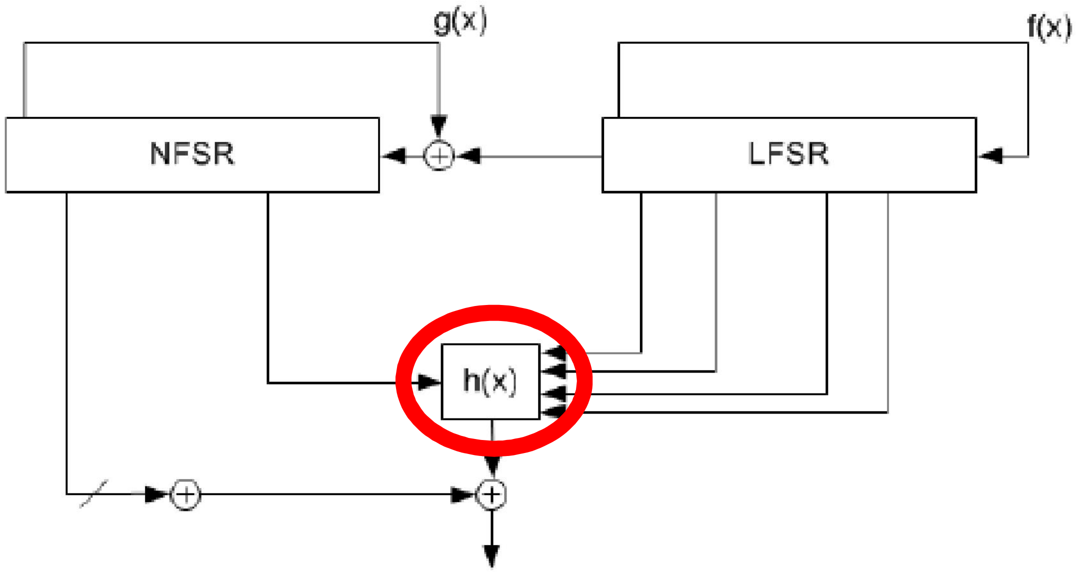


Experiments almost followed our theory!!

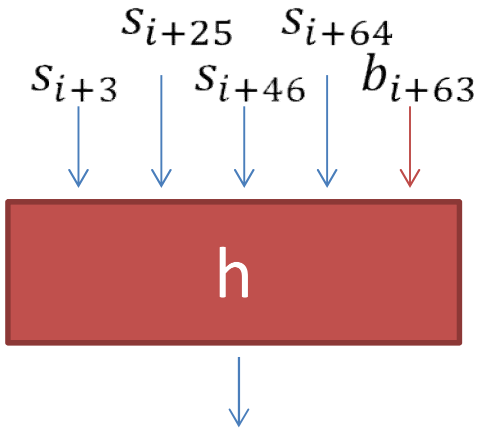
Application to Grain Family

Grain Family

- If there are many high-biased linear masks, the new framework is powerful.
- Grain-like ciphers tend to have many high-biased linear masks because of the h function.



- Example, case of Grain v1



$$h(x_0, \cdots, x_4) = x_1 + x_4 + x_0x_3 + x_2x_3 + x_3x_4 + x_0x_1x_2 + x_0x_2x_3 + x_0x_2x_4 + x_1x_2x_4 + x_2x_3x_4.$$

Correlation	Input linear mask
-2^{-2}	00011, 01001, 01010, 01011, 01101, 01111, 10110, 11000, 11011, 11100
2^{-2}	00111, 01110, 10010, 11010, 11110, 11111

Each input linear mask derives different linear approximations.

Why Grain has many approximations

- Example, case of Grain v1
  - There are 2^4 input linear mask for each active h function.

- Moreover, the sum of $|\mathbb{T}_z|$ key stream bits is used.
  - So, the potential number of approximations is $2^{4\times|\mathbb{T}_z|}$.
    - $\mathbb{T}_z = \{0,14,21,28,37,45,52,60,62,80\}$ is exploited when Grain-v1 is attacked.
    - Potentially, there are $2^{4\times 10} = 2^{40}$ different linear approximations.
  - But, in real, …
    - More complicated evaluation is required.
    - Please read our paper in detail.

Conclusion and open question

- Our attack result.

Target	# of lin. approx.	Correlations	Data	Time
Grain-128a				
Grain-128				
Grain-v1				

* For Grain-128, dynamic cube attack is more powerful.

- Open question.
  - We only break full Grain-128a w/o authentication.
  - If the authentication is enabled, only even-clock outputs are used.
    - Odd-clock ones are used for the authentication, and we cannot observe them.
    - Then, we weren’t able to find high-biased linear approximations.