# Spectral analysis of ZUC-256

5G future is here!

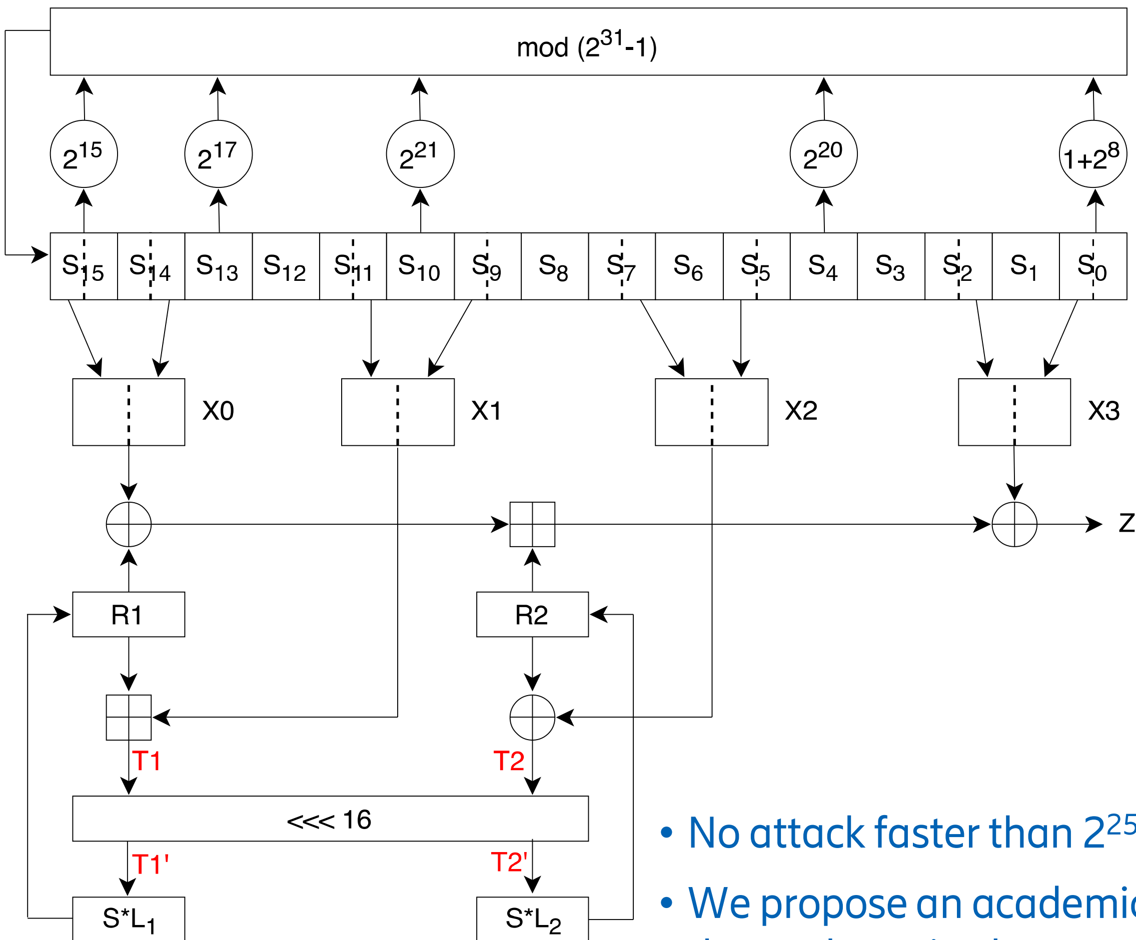

Alexander Maximov — Ericsson Research, Lund, Sweden

Jing Yang and Thomas Johansson — Lund University, Lund, Sweden


- The algorithm of ZUC-256
- Attack approaches
- Spectral analysis tools


Fast Software Encryption 2020, November 9-13

# Introduction of ZUC-128/256




- Domestic cipher used in China
- 32-bit oriented stream cipher
- FSM over $GF(2^{32})$
- LFSR over prime modulo $p=2^{31}-1$
- BR layer

- [2011] 3GPP standard UEA3/UIA3 with 128-bit key
- [2018] ZUC-256 was proposed as a 256-bit key version for 5G air encryption
  - *Eurocrypt 2018 Rump session*
  - *ZUC-256 Workshop*

- No attack faster than $2^{256}$ found (until now)
- We propose an academic attack $2^{20}$ faster than exhaustive key search

# Linear approximation: $Z_p \rightarrow 2\text{xGF}(2^{16})$

- Start from the LFSR and BR layer

$$s^{(t_1)} + s^{(t_2)} = s^{(t_3)} + s^{(t_4)} \mod p \qquad (1)$$

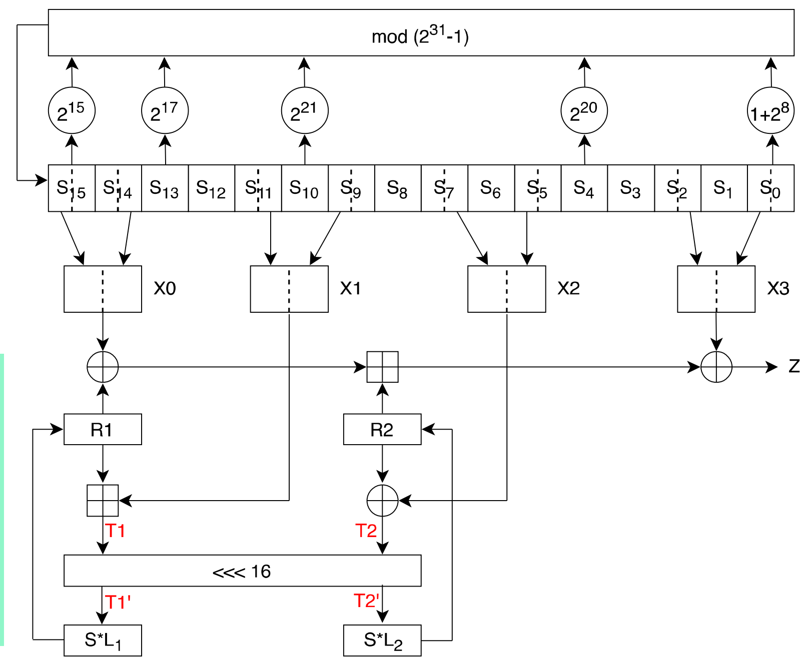
- Approximate as $2\text{xGF}(2^{16})$

$$X^{(t_1)} \boxplus_{16} X^{(t_2)} = X^{(t_3)} \boxplus_{16} X^{(t_4)} \boxplus_{16} C^{(t_1)}$$

- Example: for $X^{(t_i)} = X1^{(t_i)}$

$$\underbrace{\begin{pmatrix} s_H^{(t_1+9)} \\ s_L^{(t_1+11)} \end{pmatrix}}_{X1^{(t_1)}} \boxplus_{16} \underbrace{\begin{pmatrix} s_H^{(t_2+9)} \\ s_L^{(t_2+11)} \end{pmatrix}}_{X1^{(t_2)}} = \underbrace{\begin{pmatrix} s_H^{(t_3+9)} \\ s_L^{(t_3+11)} \end{pmatrix}}_{X1^{(t_3)}} \boxplus_{16} \underbrace{\begin{pmatrix} s_H^{(t_4+9)} \\ s_L^{(t_4+11)} \end{pmatrix}}_{X1^{(t_4)}} \boxplus_{16} \underbrace{\begin{pmatrix} C1_L^{(t_1)} \\ C1_H^{(t_1)} \end{pmatrix}}_{C1^{(t_1)}}$$

$$Pr\{C_L^{(t_1)} = 0\} = Pr\{C_H^{(t_1)} = 0\} \approx 2/3$$
$$Pr\{C_L^{(t_1)} = -1\} = Pr\{C_H^{(t_1)} = -1\} \approx 1/6$$
$$Pr\{C_L^{(t_1)} = +1\} = Pr\{C_H^{(t_1)} = +1\} \approx 1/6$$



# Linear approximation: Deriving biased samples

$$X^{(t_1)} \boxplus_{16} X^{(t_2)} = X^{(t_3)} \boxplus_{16} X^{(t_4)} \boxplus_{16} C^{(t_1)}$$

- Two consecutive keystream words

$$Z^{(t)} = [(T2^{(t)} \oplus X2^{(t)}) \boxplus ((T1^{(t)} \boxminus X1^{(t)}) \oplus X0^{(t)})] \oplus X3^{(t)}$$

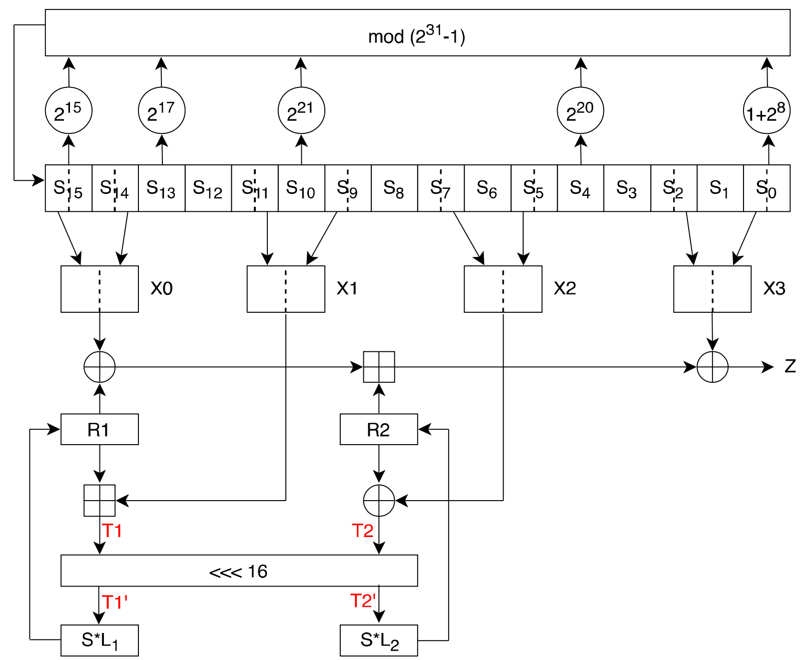
$$Z^{(t+1)} = [SL_2(T2'^{(t)}) \boxplus (SL_1(T1'^{(t)}) \oplus X0^{(t+1)})] \oplus X3^{(t+1)}$$

- **New idea:** Include LFSR cancellation into the full noise expression, thus making the bias larger

$$M\sigma[Z^{(t_1)} \oplus Z^{(t_2)} \oplus Z^{(t_3)} \oplus Z^{(t_4)}] \oplus [Z^{(t_1+1)} \oplus Z^{(t_2+1)} \oplus Z^{(t_3+1)} \oplus Z^{(t_4+1)}]$$

$$= M\sigma N1^{(t_1)} \oplus N2^{(t_1)}$$

$$\oplus \bigoplus_{t \in \{t_1,\ldots,t_4\}} \left[M \cdot T1'^{(t)} \oplus SL_1(T1'^{(t)}) \oplus M \cdot T2'^{(t)} \oplus SL_2(T2'^{(t)})\right]$$




- σ – swap of high and low 16 bits
- *M* – 32x32 Boolean matrix that the attacker can choose

# Academic distinguishing attack: Results

- **Sampling**

$$M\sigma[Z^{(t_1)} \oplus Z^{(t_2)} \oplus Z^{(t_3)} \oplus Z^{(t_4)}] \oplus [Z^{(t_1+1)} \oplus Z^{(t_2+1)} \oplus Z^{(t_3+1)} \oplus Z^{(t_4+1)}]$$

- **Total noise expression (details on N1 and N2 will be given later)**

$$= M\sigma N1^{(t_1)} \oplus N2^{(t_1)}$$

$$\oplus \bigoplus_{t\in\{t_1,\ldots,t_4\}} \left[M \cdot T1'^{(t)} \oplus SL_1(T1'^{(t)}) \oplus M \cdot T2'^{(t)} \oplus SL_2(T2'^{(t)})\right]$$

- **Found matrix M**

```
uint32_t M[32] =
{  0x26dad00b, 0x5de94454, 0x3bdfdb0d, 0x1423c42f, 0xc4f35585, 0x1f22e504,
   0xeb07cc1e, 0x3633b301, 0x11b4bca3, 0x6f23b103, 0x912adb7d, 0x6a058e9e,
   0x67d4ef5a, 0xdd0830b6, 0xee579099, 0x9af30192, 0x455d8a7b, 0x22133144,
   0x7fb935a8, 0x4d923b96, 0xc0c9967e, 0x99db94fc, 0x442f1154, 0x17994e1f,
   0x08d2662e, 0xccc8fe9c, 0x994d8fb8, 0xfba4f0dc, 0x462d2a69, 0x373306ed,
   0x91282e11, 0x9b82d788 };
```

- **Bias of the total noise (Squared Euclidean Imbalance, SEI)**

$$\epsilon(N_{tot}^{(t_1)}) \approx 2^{-236.380623}$$

- **Distinguishing attack complexity is O(1/ε) = O($2^{236}$)**
  - in $s^{(t_1)} + s^{(t_2)} = s^{(t_3)} + s^{(t_4)} \mod p$ the degree is ~$2^{167}$

- **Problem 1:**
  - Computation of 32-bit noise distributions *(adapted "bit-slicing" technique)*
- **Problem 2:**
  - Searching for the 32x32 binary masking matrix M *(spectral analysis)*

# Noise expressions and "Bit-slicing" technique

$$X^{(t_1)} \boxplus_{16} X^{(t_2)} = X^{(t_3)} \boxplus_{16} X^{(t_4)} \boxplus_{16} C^{(t_1)}$$

$$\begin{aligned} N1a^{(t_1)} = & [((T2^{(t_1)} \oplus X2^{(t_1)}) \boxplus ((T1^{(t_1)} \boxminus X1^{(t_1)}) \oplus X0^{(t_1)}))] \\ & \oplus [((T2^{(t_2)} \oplus X2^{(t_2)}) \boxplus ((T1^{(t_2)} \boxminus X1^{(t_2)}) \oplus X0^{(t_2)}))] \\ & \oplus [((T2^{(t_3)} \oplus X2^{(t_3)}) \boxplus ((T1^{(t_3)} \boxminus X1^{(t_3)}) \oplus X0^{(t_3)}))] \\ & \oplus [((T2^{(t_4)} \oplus (X2^{(t_1)} \boxplus_{16} X2^{(t_2)} \boxminus_{16} X2^{(t_3)} \boxminus_{16} C2^{(t_1)})) \boxplus ((T1^{(t_4)} \\ & \boxminus (X1^{(t_1)} \boxplus_{16} X1^{(t_2)} \boxminus_{16} X1^{(t_3)} \boxminus_{16} C1^{(t_1)})) \\ & \oplus (X0^{(t_1)} \boxplus_{16} X0^{(t_2)} \boxminus_{16} X0^{(t_3)} \boxminus_{16} C0^{(t_1)})))] \oplus \bigoplus_{t \in \{t_1, \ldots, t_4\}} (T1^{(t)} \oplus T2^{(t)}) \end{aligned}$$

$$N1b^{(t_1)} = X3^{(t_1)} \oplus X3^{(t_2)} \oplus X3^{(t_3)} \oplus (X3^{(t_1)} \boxplus_{16} X3^{(t_2)} \boxminus_{16} X3^{(t_3)} \boxminus_{16} C3^{(t_1)})$$

$$\begin{aligned} N2^{(t_1)} = & [[(SL_2(T2'^{(t_1)}) \boxplus (SL_1(T1'^{(t_1)}) \oplus X0^{(t_1+1)})) \oplus X3^{(t_1+1)}] \\ & \oplus [(SL_2(T2'^{(t_2)}) \boxplus (SL_1(T1'^{(t_2)}) \oplus X0^{(t_2+1)})) \oplus X3^{(t_2+1)}] \\ & \oplus [(SL_2(T2'^{(t_3)}) \boxplus (SL_1(T1'^{(t_3)}) \oplus X0^{(t_3+1)})) \oplus X3^{(t_3+1)}] \\ & \oplus [(SL_2(T2'^{(t_4)}) \boxplus (SL_1(T1'^{(t_4)}) \oplus (X0^{(t_1+1)} \boxplus_{16} X0^{(t_2+1)} \\ & \boxminus_{16} X0^{(t_3+1)} \boxminus_{16} C0^{(t_1+1)}))) \oplus (X3^{(t_1+1)} \boxplus_{16} X3^{(t_2+1)} \boxminus_{16} X3^{(t_3+1)} \\ & \boxminus_{16} C3^{(t_1+1)})]] \oplus \bigoplus_{t \in \{t_1, \ldots, t_4\}} (SL_1(T1'^{(t)}) \oplus SL_2(T2'^{(t)})) \end{aligned}$$

- Problem:
  - 32-bit noise variables
  - Just computing Dist(N1a) would require a loop of size $9^3 * 2^{17*32}$ !
- Solution:
  - Compute with adapted "Bit-slicing" technique in time $\sim O(2^{47})$.

# Problem 2: Searching for the linear masking matrix M

- Recall the total noise expression:

$$N_{tot}^{(t_1)} = M\sigma N1^{(t_1)} \oplus N2^{(t_1)}$$
$$\oplus \bigoplus_{t\in\{t_1,\ldots,t_4\}} \left[SL_1(T1'^{(t)}) \oplus M \cdot T1'^{(t)} \oplus SL_2(T2'^{(t)}) \oplus M \cdot T2'^{(t)}\right]$$

- Assume we have computed the distributions of 32-bit noise variables N1 and N2.

- **Problem:** How to find a good 32x32 binary matrix M and to maximize the total bias?

- **Solution:** Spectral analysis techniques (next slides)

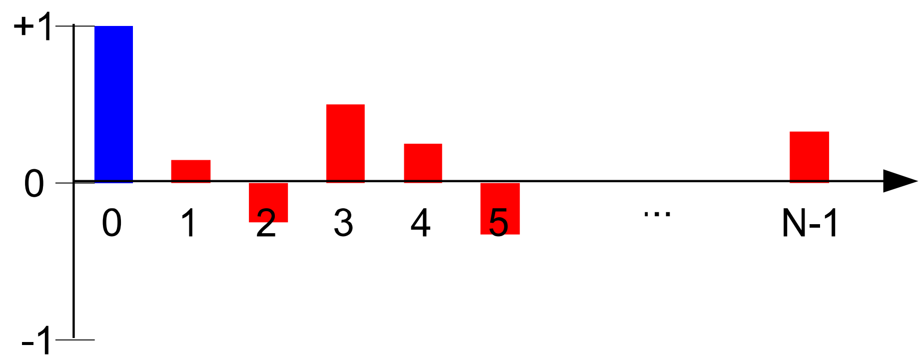
# Spectral tools: Introduction

- n-bit variables, size of the alphabet $N = 2^n$
- t- random variables (noise variables) $X^{(1)}, X^{(2)}, \ldots, X^{(t)}$
- For a random variable X, individual values are $X_0, X_1, \ldots, X_{N-1}$
- WHT and DFT $\mathcal{W}(X)_k$ and $\mathcal{F}(X)_k$, for $k = 0, 1, \ldots, N-1$

$$\hat{X}_k = \mathcal{F}(X)_k = \sum_{j=0}^{N-1} X_j \cdot e^{-\frac{i2\pi}{N}kj}$$

$$\hat{X}_k = \mathcal{W}(X)_k = \sum_{j=0}^{N-1} X_j \cdot (-1)^{k \cdot j}$$

- What can we do in frequency domain for cryptanalysis?
  - Bias computation and precision problem
  - Convolutions of noise distributions
  - Search for a linear masking (e.g. nxn binary matrix M)
  - Approximation of S-Boxes
  - ...etc




$$f = |\hat{X}_0| = \sum_{j=0}^{N-1} X_j$$

# Spectral tools: Bias computation and precision problem

- Bias = Squared Euclidean Imbalance ($f$ = normalization factor)

$$\epsilon(X) = N \sum_{i=0}^{N-1} (X_i/f - 1/N)^2$$

  - A distinguisher needs $O(1/\epsilon(X))$ samples

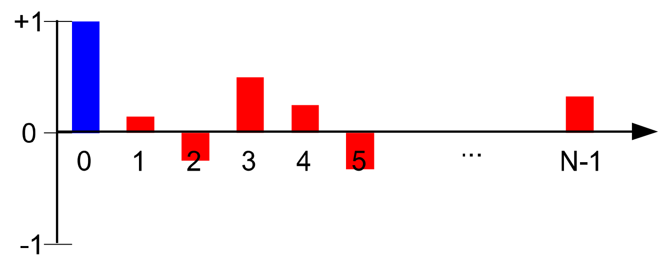
- **Theorem 1:** bias computation in the frequency domain

$$\epsilon(X) = \frac{1}{|\hat{X}_0|^2} \sum_{i=1}^{N-1} |\hat{X}_i|^2$$

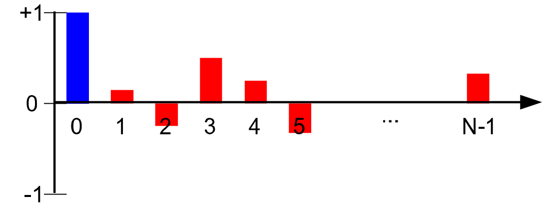
**Consequences**

- In the frequency domain only low precision is needed, but with the exponent field
- Data type **double** in standard C is good enough (exponent value up to $2^{-1023}$)
- Works even if the initial distribution of X is not normalized (then f is used)

- **Problem:** if expected bias is ~$2^{-p}$ then in time domain the values must have precision at least O(|p/2|) bits!
  - Example: for an expected bias $2^{-512}$ we must handle large number arithmetic and have precision >256 bits.



# Spectral tools: Convolutions




- From e.g. [MJ05]

$$(X^{(1)} \boxplus X^{(2)} \boxplus \ldots \boxplus X^{(t)}) = \mathcal{F}^{-1}(\mathcal{F}(X^{(1)}) \cdot \mathcal{F}(X^{(2)}) \cdot \ldots \cdot \mathcal{F}(X^{(t)}))$$

$$(X^{(1)} \oplus X^{(2)} \oplus \ldots \oplus X^{(t)}) = \mathcal{W}^{-1}(\mathcal{W}(X^{(1)}) \cdot \mathcal{W}(X^{(2)}) \cdot \ldots \cdot \mathcal{W}(X^{(t)}))$$

- Consequence: the bias of a convolution

$$\epsilon(X^{(1)} \boxplus \ldots \boxplus X^{(t)}) = \frac{1}{f} \sum_{k=1}^{N-1} |\mathcal{F}(X^{(1)})_k|^2 \cdot \ldots \cdot |\mathcal{F}(X^{(t)})_k|^2 = \frac{1}{f} \sum_{k=1}^{N-1} \left( \prod_{i=1}^{t} |\mathcal{F}(X^{(i)})_k| \right)^2,$$

$$\text{where} \quad f = |\mathcal{F}(X^{(1)})_0|^2 \cdot \ldots \cdot |\mathcal{F}(X^{(t)})_0|^2 = \left( \prod_{i=1}^{t} |\mathcal{F}(X^{(i)})_0| \right)^2$$

**Observation & Motivation**

- Peak spectrum values contribute the most to the total bias
- Motivates to learn how to “shuffle” spectrums by some manipulations in the time domain.

# Spectral tools: Linear masking (WHT case)

- Given $t$ noise distributions $X^{(q)}, q = 0, 1, \ldots, t$, find $t$ $n \times n$ full-rank Boolean matrices $M^{(q)}$ that maximize $n$ spectral points of $X$ in the expression:

$$X = M^{(1)}X^{(1)} \oplus M^{(2)}X^{(2)} \oplus \ldots \oplus M^{(t)}X^{(t)}$$

- **Theorem 2:** $\mathcal{W}(M \cdot X)_k = \mathcal{W}(X)_{k \cdot M}$

- **Algorithm 1: (solution to find M-matrices above)**
  - Place wanted n indexes as rows of the $n \times n$ matrix $K$ (must be full rank)
  - For each $X^{(q)}$ find n spectral indexes with peak spectral values (sorted descending order). Place those indexes as rows of $\Lambda^{(q)}$ (must be full rank)
  - Derive $M^{(q)} = K^{-1} \cdot \Lambda^{(q)}$

$$\mathcal{W}(M^{(q)} \cdot X^{(q)})_{k_0} = \mathcal{W}(X^{(q)})_{k_0 \cdot M^{(q)}} = \mathcal{W}(X^{(q)})_{\lambda_0^{(q)}} \to \text{peak}$$

# Spectral tools: Linear masking (DFT case)

- Given $t$ noise distributions $X^{(i)}, i = 0, 1, \ldots, t$, find $t$ odd constants $c_i$ that maximize the peak spectrum value of $X$ in the expression:

$$X = c_1 X^{(1)} \boxplus c_2 X^{(2)} \boxplus \ldots \boxplus c_t X^{(t)}$$

- **Theorem 6:** $\mathcal{F}(c \cdot X)_k = \mathcal{F}(X)_{k \cdot c} \mod N$

- **Cor. 2&3:** $\mathcal{F}(X)_{\underbrace{2^m(1+2q)}_{=k}} = \mathcal{F}(\underbrace{(1+2q)}_{=c} \cdot X)_{2^m}$

- **Algorithm 3: (solution to find c-constants above)**
  - Locate the "group" m where the maximum peak value is happening over the product of group-max values for all Xs
  - Set $c_i$ such that it "rotates" the corresponding spectrum within the group m
  - Best alignment happens at the point $2^m$

# Spectral tools: Approximation of S-Boxes (Intro)

- Examples for composite S-Box constructions:

$$\underbrace{\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{matrix} S_0( \\ S_1( \\ S_0( \\ S_1( \end{matrix} \begin{pmatrix} L_1 \\ 32 \times 32 \\ \text{binary} \\ \text{matrix} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} \begin{matrix} ) \\ ) \\ ) \\ ) \end{matrix}}_{\text{used in ZUC}} \qquad \underbrace{\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{pmatrix} \cdot \begin{pmatrix} S_R(w_0) \\ S_R(w_1) \\ S_R(w_2) \\ S_R(w_3) \end{pmatrix}}_{\text{used in SNOW-3G}}$$

- Example of an approximation: $X = RS(Qx) \oplus Mx$

- **Questions:**
  - How to find M such that the bias of X is large?
  - How to derive the spectrum value of X at index k?

# Spectral tools: Usual S-Boxes

- For an $n$-bit S-box $S(x)$ and an $n$-bit integer $k$ define the $k$-th binary-valued (i.e., $\pm 1/N$) function:

$$B^{[k]}_{\{S(x)\}} = 1/N \cdot (-1)^{k \cdot S(x)}, \quad \text{for } x = 0, 1, \ldots, N-1$$

- **Theorem 3:** $\mathcal{W}(S(x) \oplus M \cdot x)_k = \mathcal{W}(B^{[k]}_{\{S(x)\}})_{k \cdot M}$

- **Algorithm 2: (Find a good masking matrix M)**
  - for each k>0 compute WHT: $\mathcal{W}(B^{[k]}_{\{S(x)\}})$
  - loop for λ-index over the k-th spectrum above
  - collect many enough triples

    $$\{(k, \lambda, \omega)\} : \ \omega = \left|\mathcal{W}(B^{[k]}_{\{S(x)\}})_\lambda\right| \to \max$$
  - from the triples $\{(k, \lambda, \omega)\}$ construct full-rank matrices $K$ and $\Lambda$ with greedy approach
  - derive $M = K^{-1}\Lambda$

# Spectral tools: Composite S-Boxes

$$B^{[k]}_{\{S(x)\}} = 1/N \cdot (-1)^{k \cdot S(x)}, \quad \text{for } x = 0, 1, \ldots, N-1$$

- **Theorem 5:** If $n$-bit S-box is constructed from $t$ smaller $n_1, n_2, \ldots, n_t$-bit S-boxes: $S(x) = \begin{pmatrix} S_1(x_1) & S_2(x_2) & \ldots & S_t(x_t) \end{pmatrix}^{\mathrm{T}}$ then

$$\mathcal{W}(B^{[k]}_{\{S(x)\}})_\lambda = \prod_{i=1}^{t} \mathcal{W}(B^{[k_i]}_{\{S_i(x)\}})_{\lambda_i}.$$

where $x = (x_1|x_2|\ldots|x_t)$, $k = (k_1|k_2|\ldots|k_t)$, $\lambda = (\lambda_1|\lambda_2|\ldots|\lambda_t)$.

- **Usage example:**
  - for all basic S-Boxes (8-bit S0/S1 in ZUC) precompute tables like $T_i[k_i, \lambda_i] = \mathcal{W}(B^{[k_i]}_{\{S_i(x)\}})_{\lambda_i}$
  - then any spectrum values of a large composite S-Box can be derived through these tables:

let $X = RS(Qx) \oplus Mx$, then for any $k$ compute $\lambda = k \cdot M,\ k' = k \cdot R,\ \lambda' = \lambda \cdot Q^{-1}$

$$\mathcal{W}(X)_k = \prod_{i=1}^{t} \mathcal{W}(B^{[k'_i]}_{\{S_i(x)\}})_{\lambda'_i} = \prod_{i=1}^{t} T_i[k'_i, \lambda'_i]$$

# Spectral analysis of ZUC – the final step!

- Recall the total noise expression:

$$N_{tot}^{(t_1)} = M\sigma N1^{(t_1)} \oplus N2^{(t_1)} \oplus \bigoplus_{t\in\{t_1,\ldots,t_4\}} \left[SL_1(T1'^{(t)}) \oplus M\cdot T1'^{(t)} \oplus SL_2(T2'^{(t)}) \oplus M\cdot T2'^{(t)}\right]$$

- For any point k, the spectral expression for the total noise:

$$\mathcal{W}(N_{tot}^{(t_1)})_k = \mathcal{W}(M\sigma N1)_k \cdot \mathcal{W}(N2)_k \cdot \mathcal{W}(SL_1(x)\oplus Mx)_k^4 \cdot \mathcal{W}(SL_2(x)\oplus Mx)_k^4$$
$$= \mathcal{W}(\sigma N1)_\lambda \cdot \mathcal{W}(N2)_k \cdot \mathcal{W}(B_{\{SL_1(x)\}}^{[k]})_\lambda^4 \cdot \mathcal{W}(B_{\{SL_2(x)\}}^{[k]})_\lambda^4,$$

where $\lambda = k\cdot M$.

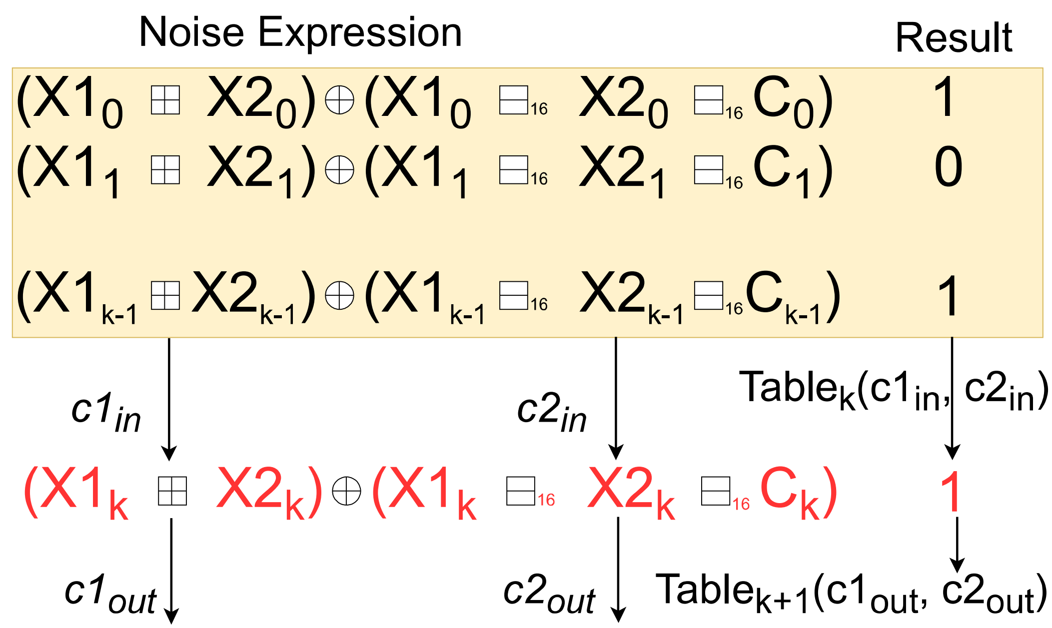
- **Spectral analysis of ZUC: our strategy for the final step to find M**
  - we selected ~$2^{24.78}$ "promising" λ-points where $|\mathcal{W}(\sigma N1)_\lambda|^2 > 2^{-150}$
  - we selected ~$2^{18}$ "promising" k-points where $|\mathcal{W}(N2)_k|^2 > 2^{-80}$
  - for each pair (k, λ) we compute the spectrum value, then collect best pairs (k, λ)
  - construct matrices $K$ and $\Lambda$ and derive $M = K^{-1}\cdot\Lambda$

# Bit-slicing technique: Basics

- *N1a, N1b, N2* are 32-bit noise variables:
  - have 32-bit operators $\oplus, \boxplus, \boxminus$
  - 2x16-bit operators $\boxplus_{16}, \boxminus_{16}$
  - the carry random variables $C = \{0, -1, +1\}$.

- Consider a 32-bit "toy" noise expression *N* (we use the same techniques to compute N1a, N1b, N2).

$$N = (X1 \boxplus X2) \oplus (X1 \boxminus_{16} X2 \boxminus_{16} C)$$

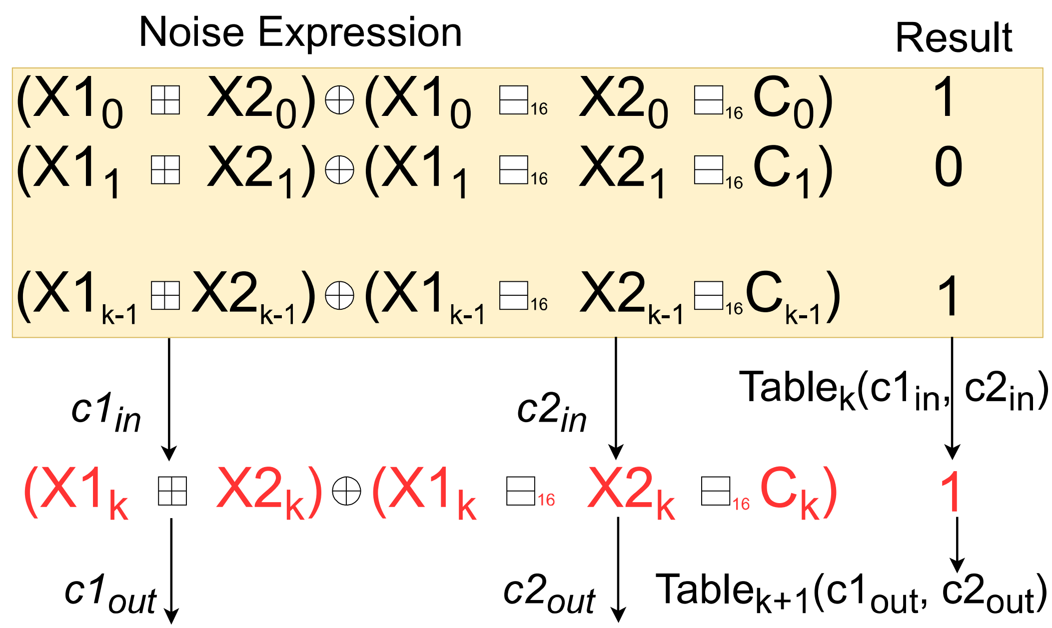



- **Table$_k$(c1, c2...)** = number of combinations of k-bit truncated input variables (X1, X2...) such that the result is a wanted k-bit truncated result **R** and the output sub-carries are **c1** and **c2**.
- Given **Table$_k$(c1, c2...)** and $r_k$ it is easy to compute **Table$_{k+1}$(c1, c2...)**
- Transition from k'th table to (k+1)'th is a linear operation => transition matrices **$M_x$**, where $x=r_k$.
- **Table$_k$(c1, c2...)** ➔ vector **$V_k$** of length **t**.

# Bit-slicing technique: Basics

- Two transition matrices can be precomputed:

$$M_0 \text{ and } M_1$$




- General formulae:

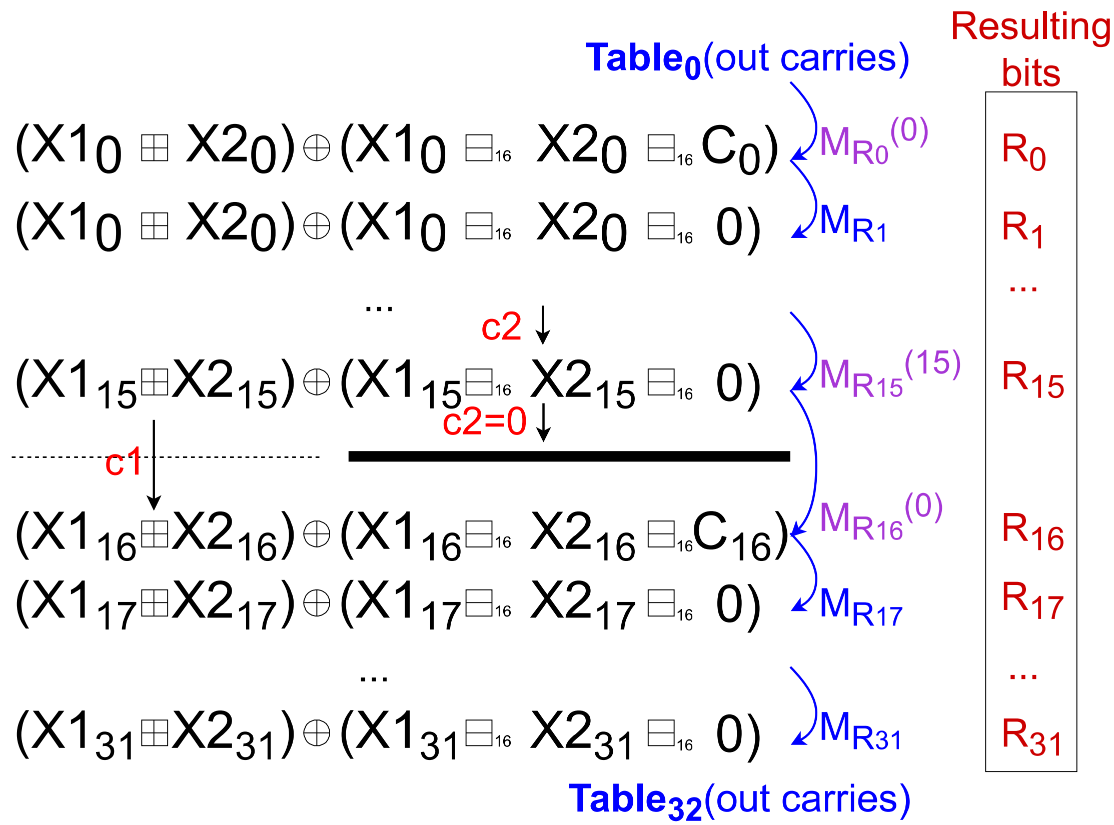
$$\Pr\{N = (r_{n-1} \dots r_0)\} = \frac{1}{2^{t \cdot n}} \cdot \underbrace{(1, 1, \dots, 1) \cdot \prod_{i=n/2}^{n-1} M_{r_i}}_{\text{High part, } H[(r_{n-1} \dots r_{n/2})]} \cdot \underbrace{\prod_{i=0}^{n/2-1} M_{r_i} \cdot V_0}_{\text{Low part, } L[(r_{n/2-1} \dots r_0)]}$$

- Precomputation of high and low parts.

# Bit-slicing technique: Adaptation

$$\Pr\{N = (r_{n-1} \dots r_0)\} = \frac{1}{2^{t \cdot n}} \cdot (1, 1, \dots, 1) \cdot \prod_{i=n/2}^{n-1} M_{r_i} \cdot \prod_{i=0}^{n/2-1} M_{r_i} \cdot V_0$$

- $C_0$ and $C_{16}$ are independent variables in range {0, -1, +1} with certain probabilities.
  - Table's entries are #of combinations * Pr{$C_0$, $C_{16}$}
- Special transition matrices for bits 0, 15, 16




- Transition matrices are of size $2^{12.8}$x$2^{12.8}$ (365Mb of RAM each)
- L/H vectors:
  - truncated lengths $t=2^8$.
  - precomputation time $O(2^{46.6})$

Two consecutive words of ZUC, at some time $t$, are expressed as:

$$\begin{aligned} Z^{(t)} &= [(T2^{(t)} \oplus X2^{(t)}) \boxplus ((T1^{(t)} \boxminus X1^{(t)}) \oplus X0^{(t)})] \oplus X3^{(t)}, \\ Z^{(t+1)} &= [SL_2(T2'^{(t)}) \boxplus (SL_1(T1'^{(t)}) \oplus X0^{(t+1)})] \oplus X3^{(t+1)}, \end{aligned}$$

In our approximation of the FSM part we basically do:

$$\begin{aligned} M\sigma Z^{(t)} \oplus Z^{(t+1)} &= M\sigma[[(T2^{(t)} \oplus X2^{(t)}) \boxplus ((T1^{(t)} \boxminus X1^{(t)}) \oplus X0^{(t)})] \oplus X3^{(t)}] \\ &\oplus [SL_2(T2'^{(t)}) \boxplus (SL_1(T1'^{(t)}) \oplus X0^{(t+1)})] \oplus X3^{(t+1)} \\ &= M\sigma[N1^{(t)} \oplus T2^{(t)} \oplus X2^{(t)} \oplus T1^{(t)} \oplus X1^{(t)} \oplus X0^{(t)} \oplus X3^{(t)}] \\ &\oplus N2^{(t)} \oplus SL_2(T2'^{(t)}) \oplus SL_1(T1'^{(t)}) \oplus X0^{(t+1)} \oplus X3^{(t+1)} \\ &= M\sigma N1^{(t)} \oplus N2^{(t)} \\ &\oplus M\sigma(X2^{(t)} \oplus X1^{(t)} \oplus X0^{(t)} \oplus X3^{(t)}) \oplus X0^{(t+1)} \oplus X3^{(t+1)} \\ &\oplus M(\underbrace{\sigma T2^{(t)} \oplus \sigma T1^{(t)}}_{=T2'^{(t)} \oplus T1'^{(t)}}) \oplus SL_2(T2'^{(t)}) \oplus SL_1(T1'^{(t)}) \end{aligned}$$

Thus we get the following:

$$\begin{aligned} M\sigma Z^{(t)} \oplus Z^{(t+1)} &= M\sigma N1^{(t)} \oplus N2^{(t)} - \text{ noise variables from approximations of } \boxplus, \boxminus\text{s to } \oplus\text{s} \\ &\oplus \quad \underbrace{M\sigma(X2^{(t)} \oplus X1^{(t)} \oplus X0^{(t)} \oplus X3^{(t)}) \oplus X0^{(t+1)} \oplus X3^{(t+1)}}_{\text{These X-terms to be cancelled by adding the above FSM approx in 4 time instances}} \\ &\oplus \underbrace{M \cdot T2'^{(t)} \oplus SL_2(T2'^{(t)}) \oplus M \cdot T1'^{(t)} \oplus SL_1(T1'^{(t)})}_{\text{These are just another noise terms, seen as S-box approximations}} \end{aligned}$$