Some observations on ZUC-256 (Extended)

Alexander Maximov

Ericsson Research, Lund, Sweden
alexander.maximov@ericsson.com

Abstract. In this report we study efficient binary approximations of the FSM of ZUC-256 with high correlation around $2^{-21.1}$ between the keystream words and the LFSR. We then map these approximations into a binary distinguisher with complexity around 2^{234}. Thereafter, we convert to an approximation in the LFSR's field $\mathbb{Z}_p$ with correlation around $2^{-33.6}$. We share a number of observations and state open problems for further research and considerations.

Keywords: 5G, ZUC-256

1 Introduction

ZUC-256 [ZUC18] is a stream cipher with the target to be used in 5G as one of the 256-bit security algorithms for confidentiality and integrity. At the moment, ZUC-256 is under evaluation by ETSI SAGE and understanding it's strength is therefore important.

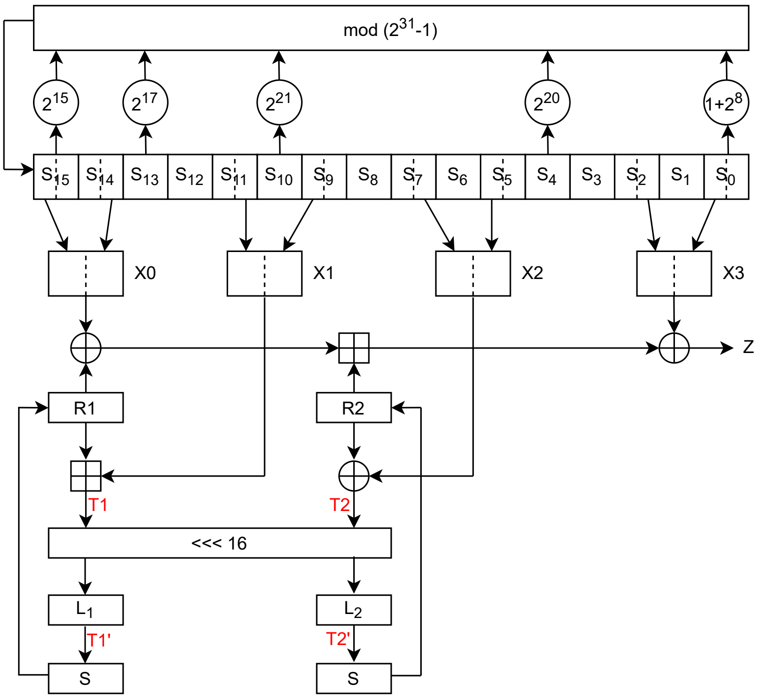

Figure 1: The keystream generation phase of the ZUC-256 stream cipher.

This work was inspired by recent results in [SJZ+21], and we decided to check whether similar methods may be applied to ZUC-256. The keystream generator of ZUC-256 is depicted on Figure 1, and for more details we refer to the original specification of the design [ZUC18].

2 FSM approximation and correlation trails

We introduce additional intermediate signals $T1, T2$ and $T1', T2'$ as on Figure 1, then two consecutive keystream words can be expressed as follows:

$$\begin{aligned} z &= (((T1 \boxminus X1) \oplus X0) \boxplus (T2 \oplus X2)) \oplus X3 \\ z' &= ((S(T1') \oplus X0') \boxplus S(T2')) \oplus X3' \\ \text{where } (T1', T2') &= (L_1, L_2) \cdot (T1, T2)_{\lll 16} = \sigma(T1, T2) \end{aligned}$$

There is a linear relation between $(T1, T2)$ and $(T1', T2')$. Thus, if we have a pair of masks (m_1, m_2) for $(T1, T2)$, then corresponding masks for $(T1', T2')$ will be (m_1', m_2') such that $(m_1, m_2) \cdot (T1, T2) = (m_1', m_2') \cdot (T1', T2')$; the masks (m_1', m_2') can be derived linearly from (m_1, m_2), and vice versa.

The steps of the FSM approximation may be derived as follows:

$$\begin{aligned} \alpha \cdot z &= \alpha \cdot [(((T1 \boxminus X1) \oplus X0) \boxplus (T2 \oplus X2)) \oplus X3] \\ &\to \underbrace{m_0 \cdot [((T1 \boxminus X1) \oplus X0] \oplus m_2 \cdot [T2 \oplus X2]}_{\rho_{\boxplus}(\alpha, m_0, m_2)} \oplus \alpha \cdot X3 \\ &\to \underbrace{m_1 \cdot T1 \oplus m_1 \cdot X1}_{\rho_{\boxminus}(m_0, m_1, m_1)} \oplus m_0 \cdot X0 \oplus m_2 \cdot T2 \oplus m_2 \cdot X2 \oplus \alpha \cdot X3 \\ \beta \cdot z' &= \beta \cdot [((S(T1') \oplus X0') \boxplus S(T2')) \oplus X3'] \\ &\to \underbrace{k_1 \cdot (S(T1') \oplus X0') \oplus k_2 \cdot S(T2')}_{\rho_{\boxplus}(\beta, k_1, k_2)} \oplus \beta \cdot X3' \\ &\to \underbrace{k_1 \cdot S(m_1' \cdot T1')}_{\rho_S(k_1, m_1')} \oplus k_1 \cdot X0' \oplus \underbrace{k_2 \cdot S(m_2' \cdot T2')}_{\rho_S(k_2, m_2')} \oplus \beta \cdot X3', \end{aligned}$$

where $\rho_{\boxplus}(s, a, b), \rho_{\boxminus}(s, a, b), \rho_S(k, m)$ are correlation values for approximations of arithmetical additions, subtractions, and S-boxes, given input and output masks. Recall that a correlation $\rho(X)$ of a random binary variable X is $\rho(X) = Pr\{X = 0\} - Pr\{X = 1\}$.

I.e., the biased binary correlation between two consecutive keystream words and the bits of the LFSR is thus expressed through X-terms, which are composed directly from the bits of the LFSR in the Bit-Reorganisation step of the cipher:

$$\alpha \cdot z \oplus \beta \cdot z' \to m_0 \cdot X0 \oplus m_1 \cdot X1 \oplus m_2 \cdot X2 \oplus \alpha \cdot X3 \oplus k_1 \cdot X0' \oplus \beta \cdot X3'.$$

For an efficient search of the masks and computation of correlation values, we utilised methods similar to [SJZ+21]. In our search for a good approximation trail with high correlation we found:

$$
\begin{array}{rlrlrl}
\alpha &= \texttt{0x01860405} & m'_1 &= \texttt{0x00040000} & \rho_{\boxplus}(\alpha, m_0, m_2) &= +2^{-10.000000} \\
m_0 &= \texttt{0x01860607} & m'_2 &= \texttt{0x00010000} & \rho_{\boxminus}(m_0, m_1, m_1) &= +2^{-4.000000} \\
m_1 &= \texttt{0x01040405} & k_1 &= \texttt{0x00300000} & \rho_{\boxplus}(\beta, k_1, k_2) &= +2^{-1.000000} \\
m_2 &= \texttt{0x01010405} & k_2 &= \texttt{0x00200000} & \rho_S(k_1, m'_1) &= -2^{-3.415037} \\
\beta &= \texttt{0x00200000} & & & \rho_S(k_2, m'_2) &= +2^{-3.192645} \\
 & & & & \rho_{tot} &= -2^{-21.607683}
\end{array}
$$

Simulation results. Because of the reality might be different from the theory, e.g., due to dependencies in the above sequence of approximations, and in order to confirm the above correlation, we run simulations of ZUC-256 and collected 2^{53} samples directly from the keystream generator. The resulting correlation value from simulations is:

$$\rho_{sim}(\text{from } 2^{53} \text{ samples}) \approx -2^{-21.093495},$$

which is a value of high confidence. These simulations confirm the found correlation, and show that it is actually slightly stronger than the theoretical one ρ_{tot}.

Platform for simulations. We have utilised a compute cloud with around 600 nodes, and 2^{53} samples were collected in about 3 days. The whole workload was split into 2^{14} sub-jobs, each initialised with a *unique* random state of ZUC-256 and producing *in average* 2^{39} samples. All the samples (or, actually, the counts) were then summed up in the end and the total of 2^{53} samples is thus received.

Relation between the two masks

We have noticed that the result we found is very similar to the correlation trail given on page 33 of [ETS11]. We did a quick simulation test of the masks from [ETS11] and after collecting 2^{48} samples we received the correlation $-2^{-21.137037}$, which is again stronger than the theoretical one $-2^{-21.6}$. I.e., at least two trails can now give the same strong correlation, which, perhaps, might be combined in some way further. The two correlation equations from this report and [ETS11] can be written as:

$$
\begin{aligned}
\texttt{0x01860405} \cdot z \oplus \texttt{0x00200000} \cdot z' \to\ & \texttt{0x01860607} \cdot X0 \oplus \texttt{0x01040405} \cdot X1 \oplus \texttt{0x01010405} \cdot X2 \\
& \oplus \texttt{0x01860405} \cdot X3 \oplus \texttt{0x00300000} \cdot X0' \oplus \texttt{0x00200000} \cdot X3' \quad \text{(in this report)} \\
\texttt{0x01040607} \cdot z \oplus \texttt{0x00200000} \cdot z' \to\ & \texttt{0x01040405} \cdot X0 \oplus \texttt{0x01860607} \cdot X1 \oplus \texttt{0x01010405} \cdot X2 \\
& \oplus \texttt{0x01040607} \cdot X3 \oplus \texttt{0x00300000} \cdot X0' \oplus \texttt{0x00200000} \cdot X3' \quad \text{(in [ETS11])}
\end{aligned}
$$

if we sum them up together, then we get the difference of the two masks:

$$\Delta: \quad \texttt{0x00820202} \cdot z \to \texttt{0x00820202} \cdot (X0 \oplus X1 \oplus X3).$$

3 Linear analysis based on the binary approximation

In [YJM19] the authors found a *32-bit multidimensional distinguisher* on ZUC-256 of complexity 2^{236} by using spectral analysis tools. Here we combine some ideas from the mentioned paper and map the found binary correlation to a *binary distinguisher*.

For random variables $s_1, s_2, s_3, s_4 \in \mathbb{Z}_p$, where $p = 2^n - 1$ is a prime, such that

$$s_1 + s_2 = s_3 + s_4 \mod p \tag{1}$$

and for a given mask τ, the following is a biased expression:

$$Q_\tau = \tau \cdot (s_1 \oplus s_2 \oplus s_3 \oplus s_4). \tag{2}$$

Observation 1. *If we assume there exist a low-degree 3-weight multiple, instead of a 4-weight as in Equation 1, then for any τ we get a very small correlation of $\rho(Q_\tau)$. I.e., it seems that a 3-weight multiple cannot be used in combination with a binary approximation of the FSM in ZUC-256, but it might be useful if an FSM approximation could be done directly over $\mathbb{Z}_p$.* □

Let $s^{(t)}$ be the value of the register S_0 in time t, then the found binary correlations can be rewritten as:

$$\alpha \cdot z^{(t)} \oplus \beta \cdot z^{(t+1)} \rightarrow \bigoplus_{i=0}^{16} \tau_i \cdot s^{(t+i)}$$

for a set of 31-bit masks $\tau_0, \ldots, \tau_{16}$, some of them are zeroes. Then, similarly to [YJM19], we find time instances (t_1, t_2, t_3, t_4), where $t_1 = 0$ and

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

which can be done with complexity 2^{167}. Finally, we derive the total expression for a sample in time t_1 as follows (it works for both masks):

$$\begin{aligned} q^{(t_1)} &= \bigoplus_{i=1}^{4} (\alpha \cdot z^{(t_i)} \oplus \beta \cdot z^{(t_i+1)}) \qquad (3)\\ &= \bigoplus_{i=1}^{4} \mu^{(t_i)} \oplus \tau_{16} s^{(t_i+16)} \oplus \tau_{15} s^{(t_i+15)} \oplus \tau_{14} s^{(t_i+14)} \oplus \tau_{11} s^{(t_i+11)} \oplus \tau_9 s^{(t_i+9)} \\ &\qquad \oplus \tau_7 s^{(t_i+7)} \oplus \tau_5 s^{(t_i+5)} \oplus \tau_3 s^{(t_i+3)} \oplus \tau_2 s^{(t_i+2)} \oplus \tau_0 s^{(t_i+0)} \end{aligned}$$

where $\mu^{(t_i)}$ is the noise variable from the binary approximation of the FSM, collected in time t_i. We know that $\rho(\mu^{(t_i)}) \approx -2^{-21.1}$, and then the bias of $q^{(t_1)}$ is

$$\rho(q^{(t_1)}) = (-2^{-21.1})^4 \cdot \prod_{i=0}^{16} \rho(Q_{\tau_i}^{(t_1)}) \tag{4}$$

By collecting about $O(\rho^{-2}(q^{(t)}))$ samples we can distinguish ZUC-256 from random. What now remains is to derive values of all involved $\rho(Q_{\tau_i}^{(t_1)})$.

Values of Q_τ for the binary correlation

Given a mask τ, we want to compute the value of $\rho(Q_\tau)$. Note that when $hw(\tau) = 1$ it follows from Theorem 7 in [YJM19] (set $t = 1$):

$$Pr\{Q_\tau = 0\} \approx 2/3 \quad \rightarrow \quad \rho(Q_\tau) \approx 1/3,$$

irrespective where the bit '1' is located in the mask τ, and the error becomes negligible as n grows. For other masks where τ is not a power of 2, we are missing the formulae.

Problem 1. *How to compute $\rho(Q_\tau)$ in Equation 2 efficiently for any inputs n and τ?* If we can find an efficient formulae to compute these values, then we could include it into the automatic search of the best FSM approximation. □

For some smaller n and p we can actually compute a full table for the noise probabilities exhaustively, for all masks τ, see Appendix A. From the study of these tables, we make the following observations, without proofs at the moment.

Observation 2. *Following the study of the tables in Appendix A we conjecture:*

1. *For any mask τ we have $\rho(Q_\tau) \geq 0$;*
2. *For any mask τ with $\rho(Q_\tau)$, and for any $k = 1..n$: $\rho(Q_{\tau \lll_n k}) = \rho(Q_\tau)$;*
3. *Let some mask $\tau = \tau_1 \oplus \tau_2$, then $\rho(Q_\tau) \geq \rho(Q_{\tau_1}) \cdot \rho(Q_{\tau_2})$.* □

We could use Observation 2(3) for an estimate of the lower bound of the sample's bias, but instead we will *simulate* these masks with $n = 31$, since we expect high correlation values there. We generate uniformly distributed random variables $s_1, s_2, s_3 \in \mathbb{Z}_p$ (i.e., $Pr\{s_i = x\} = 1/p$), then derive $s_4 \mod p$ as in Equation 1. Next, we compute the value of Q_τ as in Equation 2 and collect the statistics.

Table 1: Statistics on Q_τ for a 4-weight multiple and $n = 31$ with 2^{35} samples

	this report		[ETS11]	
τ_i	value	$\rho(Q^{(t_1)}_{\tau_i})$	value	$\rho(Q^{(t_1)}_{\tau_i})$
τ_{16}	0x00180000 → 0x003	$2^{-1.584962}$	0x00180000 → 0x003	$2^{-1.584962}$
τ_{15}	0x00c30000 → 0x0c3	$2^{-3.169244}$	0x00820000 → 0x041	$2^{-3.167130}$
τ_{14}	0x00000607 → 0x607	$2^{-4.169800}$	0x00000405 → 0x405	$2^{-4.169622}$
τ_{11}	0x00000104 → 0x041	$2^{-3.167130}$	0x00000186 → 0x0c3	$2^{-3.169244}$
τ_9	0x02028000 → 0x405	$2^{-4.169622}$	0x03038000 → 0x607	$2^{-4.169800}$
τ_7	0x00000101 → 0x101	$2^{-3.169748}$	0x00000101 → 0x101	$2^{-3.169748}$
τ_5	0x02028000 → 0x405	$2^{-4.169622}$	0x02028000 → 0x405	$2^{-4.169622}$
τ_3	0x00000020 → 0x001	$2^{-1.584962}$	0x00000020 → 0x001	$2^{-1.584962}$
τ_2	0x00000186 → 0x0c3	$2^{-3.169244}$	0x00000104 → 0x041	$2^{-3.167130}$
τ_0	0x02028000 → 0x405	$2^{-4.169622}$	0x03038000 → 0x607	$2^{-4.169800}$
	$\prod \rho(Q_\tau) \rightarrow$	$2^{-32.523996}$		$2^{-32.522020}$

Result. The total bias of the samples as in Equation 3 is

$$\rho(q^{(t_1)}) \approx 2^{-4 \cdot 21.1 - 32.5} = 2^{-116.9},$$

which leads to a distinguishing attack with complexity around $T \approx D \approx 2^{234}, M \approx 2^{167}$. The complexity appears to be similar to that in [YJM19], but here we have a *binary distinguisher.*

A hypothetical correlation attack on ZUC-256

Correlation attacks presented in [GZ21] and [SJZ+21] are quite generic for this class of stream ciphers where LFSR is involved and a biased parity check expression is available. However, the methods are only given for when the LFSR is over a field of characteristic 2, i.e., a *binary LFSR*. In ZUC-256 we, however, have to deal with a *prime LFSR* where the base field has characteristic larger than 2; in case of ZUC-256 it is the prime $p = 2^{31} - 1$. Intuitively, a correlation attack starts with a system where we have n-bits of entropy (e.g., the length of the LFSR in bits). Then we use the found correlation as a *biased binary parity check.* We then "inject" one such parity check into the system and by this the entropy of the system is reduced. By injecting many enough of such parity checks, the system becomes more and more determined, i.e., the LFSR is then recovered. We believe that the performance of such an attack for *prime LFSRs* should be similar as for *binary LFSRs.*

Problem 2. *Given a biased binary parity check on the LFSR state bits like in Equation 3, how to recover the LFSR that is over a non-binary domain $\mathbb{Z}_p$ with $p > 2$?*

At this moment we do not know an exact method how to use a *biased binary parity check* for recovering a *prime LFSR*, and we leave this as an open question. However, if the attack performance on prime LFSRs is similar to the complexity of recovering an LFSR over GF(2), then given the correlation $2^{-21.1}$ and $n \approx 496$ bits ($n \leftarrow 16 \log_2(2^{31} - 1)$), one could expect a correlation attack on ZUC-256 with complexities around $T \approx M \approx D \approx 2^{176}$, derived as in [GZ21].

4 Converting a binary approximation to a non-binary

In previous approximations we have a binary expression on the left side, $\alpha \cdot z \oplus \beta \cdot z'$, and a binary expression on the right side in terms of 31-bit words $s^{(t)}$ from the LFSR. I.e., if we take some certain bits of the LFSR state and XOR them, then we have a strong correlation to the keystream bits. We think that it might be used in a correlation attack, but at the moment we do not know an exact method, see Problem 2.

In this section we consider the possibility to convert a binary approximation into a non-binary, more specifically for the modulus $p = 2^n - 1$, where $n = 31$ is the case of ZUC-256, i.e.:

$$\alpha \cdot z \oplus \beta \cdot z' \xrightarrow{\rho=-2^{-21.1}} \tau_1 \cdot S_1 \oplus \tau_2 \cdot S_2 \oplus \dots \tag{5}$$

$$\xrightarrow{\rho=?} (\lambda_1 \cdot S_1 + \lambda_2 \cdot S_2 + \dots \mod p) \mod 2 \tag{6}$$

In this case, what we can observe on the keystream bits will correlate to the first bit of an expression on LFSR state words over $\mathbb{Z}_p$, which might lead to a more straight-forward correlation attack on ZUC-256.

Note that in this specific modulus, if $\tau = 2^k$ then it maps well to $\lambda = 2^{n-k}$, since $2^{n-k} \cdot S = S_{\ggg k}$, thus the first bits of $\tau \cdot S$ and $(\lambda \cdot S \mod p)$ will match with correlation 1. However, the situation is a bit more complicated when the binary mask τ is not just a power of 2.

We had been experimenting with these two types of approximations for smaller n, and we came to the following observations (no proofs though):

Observation 3. *Consider two 1-bit expressions on n-bit variables $S_1, S_2, \dots$ in two different domains – one is over GF(2), and another is over $\mathbb{Z}_p$, where $p = 2^n - 1$, as follows:*

$$B = \tau_1 \cdot S_1 \oplus \tau_2 \cdot S_2 \oplus \dots$$
$$A = (\lambda_1 \cdot S_1 + \lambda_2 \cdot S_2 + \dots \mod p) \mod 2$$

We are interesting in a larger correlation value between A and B, then we make the following conjectures:

1. *There may exist a high correlation between A and B if and only if the total Hamming weight of all binary masks τ_i is odd. If the total Hamming weight is even, then the maximum correlation is rather small of order around $1/p$.*

2. *Given binary masks τ_i (modulo 2^n), the masks λ_i (modulo p) for the largest overall correlation between A and B, should be constructed in the following way. Let some binary mask τ be expressed as:*

$$\tau = \sum_{t=0}^{n-1} b_t \cdot 2^t \mod 2^n, \qquad \textit{where} \;\; b_t = \{0, 1\},$$

then the best approximation A would contain a corresponding λ, constructed as

$$\lambda = \sum_{t=0}^{n-1} \pm b_t \cdot 2^{n-t} \mod p,$$

where ± indicates that there are multiple possibilities to choose from, but not all of the choices would give the largest correlation value of $B \to A$ in the full expression.

The τ-masks found in this report are *even* since the total Hamming weight is 30 bits, while the masks from [ETS11] are *odd* with 29 bits of weight. Following our discovery in Observation 3(1), only the odd binary mask can be used to approximate $B \to A$. In the following, we will proceed with searching for λ-masks given τ-masks as in [ETS11].

Problem 3 (Binary to non-binary conversion)**.** *In Observation 3, given all binary masks τs, how to find the best set of λ-masks that maximises the correlation between A and B? Given both sets of τ- and λ-masks, how to compute the correlation value efficiently?*
If we find a solution to the stated problem, then we could include it into the FSM approximation steps. □

Observation 3(2) shrinks the search space for λ-masks significantly. However, even if we have τs and construct a candidate tuple of λs, we still cannot derive the correlation value since there is no formulae. Thus, we can actually simulate and collect statistics, thus we determine the correlation that way. In order to speed up the process, we split the ten τ-masks into 3 groups (odd number), each having an odd total Hamming weight, as follows:

Group 1: $\tau_0 = \texttt{0x03038000}, \tau_2 = \texttt{0x00000104}, \tau_3 = \texttt{0x00000020}, \tau_5 = \texttt{0x02028000}$ $(hw = 11)$
Group 2: $\tau_7 = \texttt{0x00000101}, \tau_9 = \texttt{0x03038000}, \tau_{11} = \texttt{0x00000186}$ $(hw = 11)$
Group 3: $\tau_{14} = \texttt{0x00000405}, \tau_{15} = \texttt{0x00820000}, \tau_{16} = \texttt{0x00180000}$ $(hw = 7)$

I.e., for the "heaviest" groups 1 and 2 we can exhaustively test 2^{11} variants of λs, and then we perform simulations, collect about 2^{26} samples, and store those λ-tuples in each group where the sub-correlations are the largest. These simulations resulted in the following:

Table 2: Simulation results for searching of λ-tuples in each of the sub-groups.

Group	ρ	#λ-tuples
1	$\approx 2^{-4.8}$	128
2	$\approx 2^{-3.8}$	64
3	$\approx 2^{-3.2}$	32

The next step is to pick one λ-tuple in each group, and simulate the full stack of ten τ- and λ-masks. Since there the expected combined correlation is much smaller, we have to increase the number of samples to about 2^{32}, and thereafter double check with 2^{36} samples. The result is that almost every combination resulted in around $\rho(B \to A) \approx \pm 2^{-12.6}$. We give one example of a complete set of λ-masks:

$$\lambda_0 = \texttt{0x0000bfc0},\ \lambda_2 = \texttt{0x607fffff},\ \lambda_3 = \texttt{0x04000000},\ \lambda_5 = \texttt{0x0000c040}$$
$$\lambda_7 = \texttt{0x7f800000},\ \lambda_9 = \texttt{0x0000bfc0},\ \lambda_{11} = \texttt{0x20800000}$$
$$\lambda_{14} = \texttt{0x60200000},\ \lambda_{15} = \texttt{0x7fffc0ff},\ \lambda_{16} = \texttt{0x00000800}$$

simulation of which gave us the correlation $2^{-12.485698}$ after collecting 2^{36} samples.

Result. The above findings mean that an attacker can observe the first bit (with some noise) of an expression on the LFSR state in the domain over $\mathbb{Z}_p$:

$$\begin{aligned}\texttt{0x01040607} \cdot z^{(t)} \oplus \texttt{0x00200000} \cdot z^{(t+1)} \rightarrow (&\lambda_0 \cdot s^{(t)} + \lambda_2 \cdot s^{(t+2)} + \lambda_3 \cdot s^{(t+3)} \\ &+ \lambda_5 \cdot s^{(t+5)} + \lambda_7 \cdot s^{(t+7)} + \lambda_9 \cdot s^{(t+9)} + \lambda_{11} \cdot s^{(t+11)} + \lambda_{14} \cdot s^{(t+14)} \\ &+ \lambda_{15} \cdot s^{(t+15)} + \lambda_{16} \cdot s^{(t+16)} \mod p) \mod 2\end{aligned} \tag{7}$$

with the correlation:

$$\rho(\text{in modulo } p) \approx -2^{-21.1} \cdot 2^{-12.5} = -2^{-33.6}.$$

Problem 4 (FSM approximation over modulo p)**.** *How to perform a binary approximation of the FSM while extracting the halfs of the X-terms out in modulo p domain?*
In Equation 5 we basically did two steps – we first found a binary approximation, then "converted" it into an arithmetical approximation over $\mathbb{Z}_p$. However, the final correlation may be larger if we would include the arithmetical approximation into the steps of the FSM approximation directly. What we actually want to achieve is e.g.:

$$\alpha \cdot z \oplus \beta \cdot z' \oplus Noise = (c_1 \cdot X0_L + c_2 \cdot X0_H + ... + c_{12} \cdot X3'_H \mod p) \mod 2,$$

where the masks $c_i \in \mathbb{Z}_p$. That form can be mapped directly to S-terms in modulo p domain without loss of information. Moreover, in this case we might be able to use a 3-weight multiple of the LFSR feedback polynomial, which may help to reduce the complexity of a distinguishing attack significantly. □

Finally, the main reason for us to look into non-binary approximations is to get closer to the possibility of the first correlation attack on ZUC-256.

Problem 5 (Correlation attack over a non-binary field)**.** *How to use a 1-bit correlation on expressions over $\mathbb{Z}_p$, $p > 2$, like in Equation 7, for recovering the LFSR over the same field $\mathbb{Z}_p$?* □

Problem 6 (Multidimensional linear analysis)**.** *Let us have a noise expression N with mixed operations such as $\oplus, \boxplus_n$ and $+ \mod p$. How to construct an n-bit multidimensional distribution of N?*
A solution to this problem would allow us to analyse the cipher by inspecting WHT-kind of spectrum, and perform multidimensional approximations and spectral analysis of the cipher including the modulo p domain. □

References

[ETS11] ETSI SAGE. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4: Design and Evaluation Report, 2011. https://www.gsma.com/aboutus/wp-content/uploads/2014/12/EEA3_EIA3_Design_Evaluation_v2_0.pdf.

[GZ21] Xinxin Gong and Bin Zhang. Resistance of SNOW-V against fast correlation attacks. *IACR Transactions on Symmetric Cryptology*, 2021(1):378–410, Mar. 2021.

[SJZ+21] Zhen Shi, Chenhui Jin, Jiyan Zhang, Ting Cui, and Lin Ding. A correlation attack on full SNOW-V and SNOW-Vi. Cryptology ePrint Archive, Report 2021/1047, 2021. https://ia.cr/2021/1047.

[YJM19] Jing Yang, Thomas Johansson, and Alexander Maximov. Spectral analysis of ZUC-256. Cryptology ePrint Archive, Report 2019/1352, 2019. https://ia.cr/2019/1352.

[ZUC18] ZUC design team. The ZUC-256 Stream Cipher, 2018. http://www.is.cas.cn/ztzl2016/zouchongzhi/201801/W020180126529970733243.pdf.

A Exhaustive tables for $\rho(Q_\tau)$ for small $n = 5, 7$

Complete table for $n = 5, p = 31$, where $P = Pr\{Q_\tau = 0\}$:

$\tau = 00001$ $P = 0.667014$
$\tau = 00010$ $P = 0.667014$
$\tau = 00100$ $P = 0.667014$
$\tau = 01000$ $P = 0.667014$
$\tau = 10000$ $P = 0.667014$

$\tau = 00011$ $P = 0.667014$
$\tau = 00110$ $P = 0.667014$
$\tau = 01100$ $P = 0.667014$
$\tau = 11000$ $P = 0.667014$
$\tau = 10001$ $P = 0.667014$

$\tau = 00101$ $P = 0.593971$
$\tau = 01010$ $P = 0.593971$
$\tau = 10100$ $P = 0.593971$
$\tau = 01001$ $P = 0.593971$
$\tau = 10010$ $P = 0.593971$

$\tau = 00111$ $P = 0.576785$
$\tau = 01110$ $P = 0.576785$
$\tau = 11100$ $P = 0.576785$
$\tau = 11001$ $P = 0.576785$
$\tau = 10011$ $P = 0.576785$

$\tau = 01011$ $P = 0.559599$
$\tau = 10110$ $P = 0.559599$
$\tau = 01101$ $P = 0.559599$
$\tau = 11010$ $P = 0.559599$
$\tau = 10101$ $P = 0.559599$

$\tau = 01111$ $P = 0.572488$
$\tau = 11110$ $P = 0.572488$
$\tau = 11101$ $P = 0.572488$
$\tau = 11011$ $P = 0.572488$
$\tau = 10111$ $P = 0.572488$

$\tau = 11111$ $P = 0.538116$

Similar table for $n = 7, p = 127$:

$\tau = 0000001$ $P = 0.66668$
$\tau = 0000010$ $P = 0.66668$
$\tau = 0000100$ $P = 0.66668$
$\tau = 0001000$ $P = 0.66668$
$\tau = 0010000$ $P = 0.66668$
$\tau = 0100000$ $P = 0.66668$
$\tau = 1000000$ $P = 0.66668$

$\tau = 0000011$ $P = 0.66668$
$\tau = 0000110$ $P = 0.66668$
$\tau = 0001100$ $P = 0.66668$
$\tau = 0011000$ $P = 0.66668$
$\tau = 0110000$ $P = 0.66668$
$\tau = 1100000$ $P = 0.66668$
$\tau = 1000001$ $P = 0.66668$

$\tau = 0000101$ $P = 0.58545$
$\tau = 0001010$ $P = 0.58545$
$\tau = 0010100$ $P = 0.58545$
$\tau = 0101000$ $P = 0.58545$
$\tau = 1010000$ $P = 0.58545$
$\tau = 0100001$ $P = 0.58545$
$\tau = 1000010$ $P = 0.58545$

$\tau = 0000111$ $P = 0.58145$
$\tau = 0001110$ $P = 0.58145$
$\tau = 0011100$ $P = 0.58145$
$\tau = 0111000$ $P = 0.58145$
$\tau = 1110000$ $P = 0.58145$
$\tau = 1100001$ $P = 0.58145$
$\tau = 1000011$ $P = 0.58145$

$\tau = 0001001$ $P = 0.56695$
$\tau = 0010010$ $P = 0.56695$
$\tau = 0100100$ $P = 0.56695$
$\tau = 1001000$ $P = 0.56695$
$\tau = 0010001$ $P = 0.56695$
$\tau = 0100010$ $P = 0.56695$
$\tau = 1000100$ $P = 0.56695$

$\tau = 0001011$ $P = 0.56045$
$\tau = 0010110$ $P = 0.56045$
$\tau = 0101100$ $P = 0.56045$
$\tau = 1011000$ $P = 0.56045$
$\tau = 0110001$ $P = 0.56045$
$\tau = 1100010$ $P = 0.56045$
$\tau = 1000101$ $P = 0.56045$

$\tau = 0001101$ $P = 0.56045$
$\tau = 0011010$ $P = 0.56045$
$\tau = 0110100$ $P = 0.56045$
$\tau = 1101000$ $P = 0.56045$
$\tau = 1010001$ $P = 0.56045$
$\tau = 0100011$ $P = 0.56045$
$\tau = 1000110$ $P = 0.56045$

$\tau = 0001111$ $P = 0.56420$
$\tau = 0011110$ $P = 0.56420$
$\tau = 0111100$ $P = 0.56420$
$\tau = 1111000$ $P = 0.56420$
$\tau = 1110001$ $P = 0.56420$
$\tau = 1100011$ $P = 0.56420$
$\tau = 1000111$ $P = 0.56420$

$\tau = 0010011$ $P = 0.55645$
$\tau = 0100110$ $P = 0.55645$
$\tau = 1001100$ $P = 0.55645$
$\tau = 0011001$ $P = 0.55645$
$\tau = 0110010$ $P = 0.55645$
$\tau = 1100100$ $P = 0.55645$
$\tau = 1001001$ $P = 0.55645$

$\tau = 0010101$ $P = 0.53546$
$\tau = 0101010$ $P = 0.53546$
$\tau = 1010100$ $P = 0.53546$
$\tau = 0101001$ $P = 0.53546$
$\tau = 1010010$ $P = 0.53546$
$\tau = 0100101$ $P = 0.53546$
$\tau = 1001010$ $P = 0.53546$

$\tau = 0010111$ $P = 0.54071$
$\tau = 0101110$ $P = 0.54071$
$\tau = 1011100$ $P = 0.54071$
$\tau = 0111001$ $P = 0.54071$
$\tau = 1110010$ $P = 0.54071$
$\tau = 1100101$ $P = 0.54071$
$\tau = 1001011$ $P = 0.54071$

$\tau = 0011011$ $P = 0.55920$
$\tau = 0110110$ $P = 0.55920$
$\tau = 1101100$ $P = 0.55920$
$\tau = 1011001$ $P = 0.55920$
$\tau = 0110011$ $P = 0.55920$
$\tau = 1100110$ $P = 0.55920$
$\tau = 1001101$ $P = 0.55920$

$\tau = 0011101$ $P = 0.54071$
$\tau = 0111010$ $P = 0.54071$
$\tau = 1110100$ $P = 0.54071$
$\tau = 1101001$ $P = 0.54071$
$\tau = 1010011$ $P = 0.54071$
$\tau = 0100111$ $P = 0.54071$
$\tau = 1001110$ $P = 0.54071$

$\tau = 0011111$ $P = 0.53421$
$\tau = 0111110$ $P = 0.53421$
$\tau = 1111100$ $P = 0.53421$
$\tau = 1111001$ $P = 0.53421$
$\tau = 1110011$ $P = 0.53421$
$\tau = 1100111$ $P = 0.53421$
$\tau = 1001111$ $P = 0.53421$

$\tau = 0101011$ $P = 0.53571$
$\tau = 1010110$ $P = 0.53571$
$\tau = 0101101$ $P = 0.53571$
$\tau = 1011010$ $P = 0.53571$
$\tau = 0110101$ $P = 0.53571$
$\tau = 1101010$ $P = 0.53571$
$\tau = 1010101$ $P = 0.53571$

$\tau = 0101111$ $P = 0.52521$
$\tau = 1011110$ $P = 0.52521$
$\tau = 0111101$ $P = 0.52521$
$\tau = 1111010$ $P = 0.52521$
$\tau = 1110101$ $P = 0.52521$
$\tau = 1101011$ $P = 0.52521$
$\tau = 1010111$ $P = 0.52521$

$\tau = 0110111$ $P = 0.52921$
$\tau = 1101110$ $P = 0.52921$
$\tau = 1011101$ $P = 0.52921$
$\tau = 0111011$ $P = 0.52921$
$\tau = 1110110$ $P = 0.52921$
$\tau = 1101101$ $P = 0.52921$
$\tau = 1011011$ $P = 0.52921$

$\tau = 0111111$ $P = 0.52946$
$\tau = 1111110$ $P = 0.52946$
$\tau = 1111101$ $P = 0.52946$
$\tau = 1111011$ $P = 0.52946$
$\tau = 1110111$ $P = 0.52946$
$\tau = 1101111$ $P = 0.52946$
$\tau = 1011111$ $P = 0.52946$

$\tau = 1111111$ $P = 0.51796$