Date of publication SEP-30, 2021, date of current version AUG-03, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.19.3.2284

On the Statistical Analysis of ZUC, Espresso and Grain v1

SAURABH SHRIVASTAVA, K. V. LAKSHMY, CHUNGATH SRINIVASAN
TIFAC Core in Cyber Security, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
(e-mail: saurabh.srivastava@lnmiit.ac.in, kv_lakshmy@cb.amrita.edu, c_srinivasan@cb.amrita.edu)

Corresponding author: Saurabh Shrivastava (e-mail: saurabh.srivastava@lnmiit.ac.in).

ABSTRACT A stream cipher generates long keystream to be XORed with plaintext to produce ciphertext. A stream cipher is said to be secure if the keystream that it produces is consistently random. One of the ways by which we can analyze stream ciphers is by testing randomness of the keystream. The statistical tests mainly try to find if any output keystream leaks any information about the secret key or the cipher's internal state and also check the randomness of the keystream. We have applied these tests to different keystreams generated by ZUC, Espresso and Grain v1 stream ciphers to check for any weaknesses. We have also proposed four new statistical tests to analyze the internal state when the hamming weight of key and IV used is very high or low. Out of these four tests, Grain v1 fails the last test i.e. internal state correlation using high hamming weight IV.

KEYWORDS Stream cipher; Statistical test; Randomness test; Correlation; keystream.

I. INTRODUCTION

ONE Time Pad (OTP) is the perfectly secure cipher which generates truly random keystream. Plaintext is XORed with this keystream to produce ciphertext. There is no statistical weakness in truly random sequence. But OTP can't be used for encryption as it uses *n*-bit key to encrypt *n*-bit plaintext. So the key should be as long as the plaintext, which is practically impossible. So, today's stream ciphers generate a long keystream using a small fixed length key. These keystreams are pseudo-random and are not theoretically as secure as OTP but practically both are equally secure. That's why, the motive of these stream cipher is to generate pseudo-random keystreams which are not distinguishable from truly random sequences [15].

The stream ciphers take 'Key' and 'Initialization Vector (IV)' as input and generate keystreams. The length of key and IV are fixed for every stream cipher. The key is kept secret but the IV is known publically. So, the security requirement of these stream ciphers are: (a) The pseudo-random keystream should be indistinguishable from truly random sequences (b) the keystream should not reveal any information about key or cipher's initial state [11]. A weakness in stream cipher can be found at: (a) Key/IV initialization and (b) keystream generation. So, it is important to test Key/IV correlation with keystream along with testing randomness of keystream [1].

In this paper, we have applied some statistical tests such as Key/Keystream correlation test, IV/Keystream correlation test, Frame correlation test, internal state correlation test and diffusion test [6] to ZUC, Espresso and Grain v1 stream ciphers to check if keystream is leaking any information about key or IV or the internal state. SAC-r and SAC-c diffusion tests [7] have also been applied to all three ciphers. Along with that, we have checked the k-error linear complexity [9] and non-linear complexity [8] of the keystreams generated by these ciphers.

After obtaining results from these tests, we have proposed four new tests to analyze the correlation of internal states when the hamming weight of key or IV is very low or very high. These tests are internal state correlation using low hamming weight key, internal state correlation using low hamming weight IV, internal state correlation using high hamming weight key, internal state correlation using high hamming weight IV.

Organization of the paper: In the next section, a brief introduction of ciphers is given which we have analyzed. In section 3, all the tests applied to these ciphers and their results are explained. Section 4 describes about proposed tests and their results and Section 5 concludes this paper.

II. STREAM CIPHERS

A. ZUC

ZUC [18-20] is the heart of 3GPP Confidentiality and Integrity Algorithms 128-EEA3 and 128-EIA3. It is an LFSR and non-linear function based word oriented stream cipher [5]. It takes 128-bit key and 128-bit IV and generates 32-bit (a word) keystream in one cycle. It is used in 4G standard LTE [13]. Fig. 1 shows the general structure of ZUC. It has 3 logical layers: (a) Linear Feedback Shift Register (LFSR), (b) Bit Reorganization (BR) and (c) a non-linear function (F).

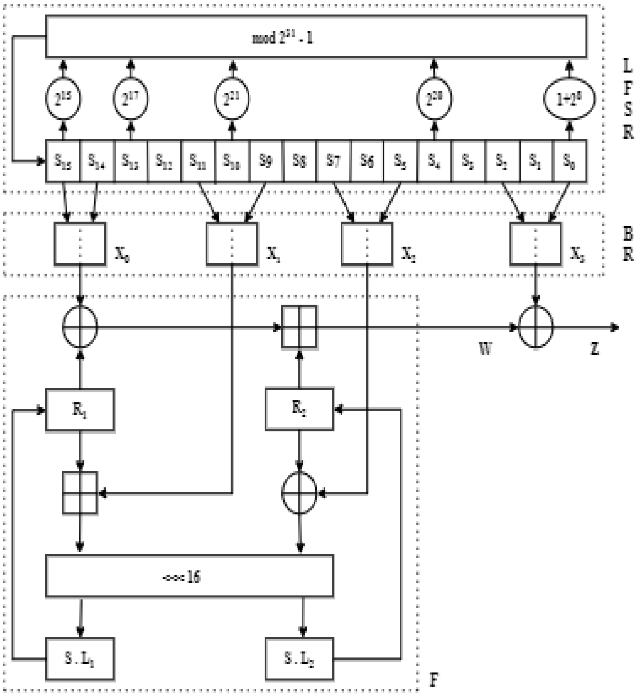

Figure 1. General structure of ZUC

B. ESPRESSO

The internal state of Espresso [17] is of 256-bit which is initialized with 128-bit initial Key, 96-bit is IV and remaining 32-bit are padded [2]. Out of 32-bit of padding, initial 31-bits are '1' and last bit is '0'. Espresso can be implemented in both Galois configuration and Fibonacci configuration. Fig. 2 shows implementation of Espresso in Fibonacci configuration.

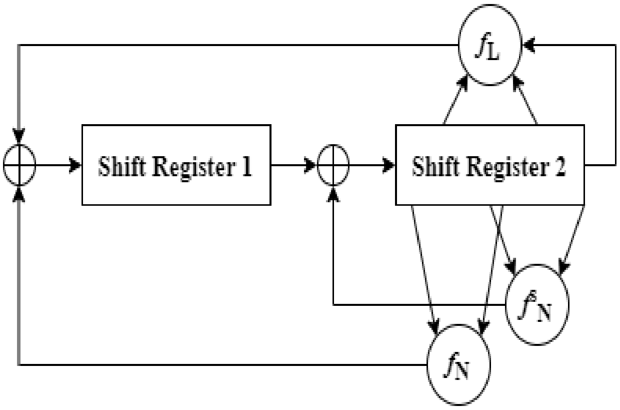

Figure 2. Espresso in Fibonacci Configuration

C. GRAIN v1

Grain v1 [16] is one of the members of Grain family of stream ciphers. Grain v0 and v1 takes 80-bit key and 64-bit IV [3]. On the other hand Grain v2 and v3 takes 128-bit key and 96-bit IV. In this paper, we are testing Grain v1 which is initialized with 160-bit where 80-bit is key, 64-bit is IV and remaining bits of internal state are padded with '1'. Fig. 3 shows the general structure of Grain family of stream cipher [4, 21].

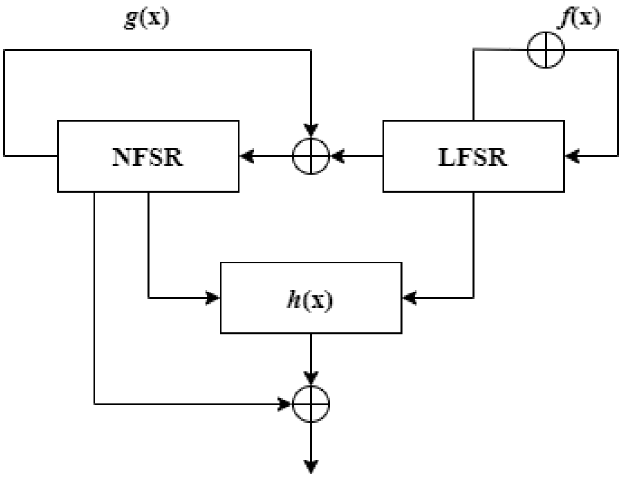

Figure 3. Overview of design blocks in Grain

III. STATISTICAL AND RANDOMNESS TESTS

In this paper we are performing five statistical tests namely Key/Keystream correlation, IV/Keystream correlation, Frame correlation, Internal State correlation and Diffusion. To analyze the result of diffusion test in more detail, SAC-r and SAC-c tests are also implemented. Along with these statistical tests, we are also performing k-error linear complexity and non-linear complexity to check the randomness of keystream.

A. KEY/KEYSTREAM CORRELATION TEST

This test is used to check the correlation between key and keystream. For this test, IV is fixed and keystream is generated for different keys where k is the length of key. We have generated N different keys and produced keystream for each key. The hamming weight is calculated after XORing keystream with respected key [6]. For a cipher to be secure, the distribution of the weights should be Binomial with parameters k and 1/2. The weight indicates the correlation between i^{th} bit of key and i^{th} bit of keystream. After this, Chi-Square Goodness of Fit test is applied and p-value is calculated. If $p \geq 0.01$, it means that there is no correlation between key and keystream with 99% surety. If the cipher fails this correlation test, there is need to revise the key initialization phase.

For this test, $N = 2^{20}$ keys are chosen randomly keeping IV constant and keystream of length k (length of key) are generated. These keystreams are XORed with respective key and stored as row of a matrix. Hamming weight of each row is calculated. The weight probability is calculated using Binomial (n, p) distribution where n = length of key and p = 1/2. The hamming weights are grouped into 5 categories with approximately equal probability as shown in Table 1.

Table 1. Weight category and probability for 128 and 80 bit

For 128-bit		For 80-bit	
Category Limit	Probability	Category Limit	Probability
0-58	0.1655	0-36	0.2170
59-62	0.2300	37-39	0.2385
63-65	0.2090	40-42	0.2562
66-69	0.2300	43-45	0.1790
70-128	0.1655	46-80	0.1093

Chi-square Goodness of Fit is applied and p-values were calculated. As we can see in Table 2, p-values for all three ciphers are greater than 0.01, there is no need to revise key initialization phase.

Table 2. Correlation between key and keystream

Ciphers	*p*-value
ZUC	0.5065
Espresso	0.2306
Grain v1	0.1099

B. IV/KEYSTREAM CORRELATION TEST

IV/Keystream correlation test is a method to find correlation between IV and Keystream. In this test, key is kept constant and N random IVs of length v are generated. For each key/IV pair, a keystream of length v is produced. The keystream is XORed with IV and hamming weight is calculated. For a cipher to be secure, these weights should be Binomial with parameter v and 1/2. After this, Chi-Square test is applied and p-value is calculated. If $p \geq 0.01$, it means that there is no correlation between IV and keystream with 99% surety. Else it means that cipher failed this correlation test and there is need to revise the IV initialization phase.

For this test, $N = 2^{20}$ IV is chosen randomly keeping the key constant. For each key/IV pair, 128-bit keystream is generated for ZUC, 96-bit keystream is generated for Espresso and 64-bit keystream is generated for Grain v1. Each keystream is XORed with respective IV and stored as row of a matrix. The hamming weight of each row is calculated. The weight probability is calculated using Binomial (n, p) distribution where n = length of IV and p = ½. The hamming weights are grouped into 5 categories with approximately equal probability as shown in Table 1 and 3.

Table 3. Weight category and probability for 96 and 64 bit

For 96-bit		For 64-bit	
Category Limit	Probability	Category Limit	Probability
0-44	0.2376	0-28	0.1909
45-47	0.2218	29-31	0.2595
48-50	0.2356	32-33	0.1957
51-53	0.1743	34-36	0.2238
54-96	0.1307	37-64	0.1302

The correlation between IV and keystream is analyzed using Chi-Square test. As you can see in Table 4, p-values for all three ciphers are greater than 0.01, which means that there is no issue with IV initialization.

Table 4. Correlation between IV and keystream

Ciphers	*p*-value
ZUC	0.5065
Espresso	0.2306
Grain v1	0.1099

C. FRAME CORRELATION TEST

For this test, a keystream of length L is generated called “Frame” using random key and IV. The aim of this test is to find the correlation between keystreams generated using constant key and similar IVs [10]. For this, key is kept constant and N-1 frames are generated by incrementing the value of IV. Using these keystreams, an $N \times L$ matrix is made and weight of each column is calculated. The distribution of these weights should be Binomial with parameters N and ½. χ^2 test is applied to check the correlation. If the test fails, there is need to revise the IV loading part.

In this test, a key/IV pair is chosen randomly and 2^{20} frames of length 1024-bit is generated by incrementing IV by 1 each time. These frames are stored as row of a matrix and column weight of each column is calculated. The weight probability is calculated using Binomial (n, p) distribution where $n = 2^{20}$ and $p = 1/2$. The hamming weights are grouped into five categories with approximately equal probability as shown in Table 5.

Table 5. Weigth category and probability for 1024 and 2^{20} bit

For 1024-bit		For 2^{20}-bit	
Category Limit	**Probability**	**Category Limit**	**Probability**
0-498	0.1994	0-523849	0.1958
499-507	0.1899	523850-524150	0.1983
508-515	0.1973	524151-524430	0.2155
516-525	0.2140	524431-524750	0.2072
526-1024	0.1994	524751-1048576	0.1832

Using χ^2 test, the correlation between frames are analyzed and as shown in Table 6, all ciphers have *p*-value greater than 0.01, that means that there is no need to revise IV loading.

Table 6. Frame correlation test

Ciphers	*p*-value
ZUC	0.9060
Espresso	0.1750
Grain v1	0.4271

D. INTERNAL STATE CORRELATION TEST

This test analyzes the effect of similar IVs on the internal state of the ciphers. This test is similar to frame correlation test. In this test, a random key/IV pair is selected and internal state ($s_1, s_2, \ldots, s_m$) is stored as a row of the matrix *M* after key/IV initialization is completed. This process is repeated *N*-1 times by increasing IV by 1 every time and thus a matrix of $N \times m$ is obtained. The weight of each column is calculated and grouped into 5 categories using Binomial distribution ($n = N, p=1/2$).

χ^2 test is applied to test the correlation of internal states. If a cipher does not pass this test, IV initialization should be revised.

The internal state of ZUC, Espresso and Grain v1 are 496, 256 and 160-bit respectively. This process is repeated $N=2^{20}$ times and matrix of $2^{20} \times m$ is created where *m* is the size of internal state. The weight of each column is calculated and grouped into 5 categories using Binomial distribution (2^{20}, 1/2).

The χ^2 test is applied to analyze the internal state correlation. As shown in Table 7, all three ciphers passed this test.

Table 7. Internal state correlation test

Ciphers	*p*-value
ZUC	0.6563
Espresso	0.9768
Grain v1	0.3949

E. DIFFUSION TEST

It is a test to examine the diffusion of every single bit of key and IV on keystream. In this test, vector $(u_1,\ldots,u_k, u_{k+1},\ldots, u_{k+v})$ is randomly selected where first *k*-bit is key and remaining *v*-bit is IV. Using this vector, a keystream is generated whose length is *L*. Then *k*+*v* vectors are obtained using operation $(u_1,\ldots,u_k, u_{k+1},\ldots, u_{k+v}) \oplus e_i$ where e_i is a vector of length *k*+*v* having 1 at i^{th} location and 0 everywhere else. Keystream is generated for each vector. Then these keystreams are XORed with the initial keystream and stored in matrix of dimension $(k + v) \times L$. Same procedure is repeated *N* times and all *N* matrices are added. Then χ^2 test is applied on each index of final matrix. If the cipher doesn't pass this test, we need to revise the initialization phase.

For this test, a key/IV pair is chosen randomly and a keystream of 1024-bit is generated. Following the above process, a matrix of $(k + v) \times 1024$ order is obtained. This process is repeated for 1024 times with different key/IV pairs and 1024 different matrices are obtained and added. The entries of matrix is grouped into 5 different categories as shown in Table 5. Now, χ^2 test is applied for analyzing the effect of diffusion. As shown in table 8, all three ciphers fail this test as *p*-value is less than 0.01 for each of these ciphers. This means that, initialization process should be revised.

Table 8. Diffusion test

Ciphers	*p*-value
ZUC	2.2111×10^{-39}
Espresso	1.3799×10^{-38}
Grain v1	4.2859×10^{-10}

F. SAC-r DIFFUSION TEST

When diffusion test fails, it means that there are some issue in initialization phase and it should be revised. But, to know the exact location where the key/IV is having weakness, we can use SAC-r and SAC-c test as proposed by Chungath Srinivasan et. al. in paper "Measuring Diffusion in Stream Ciphers using Statistical Testing Methods" [7].

In SAC-r diffusion test, a key/IV pair of length *k*+*v* is chosen and a keystream of length *L* is generated. Then the key/IV pair is XORed with vector e_i where e_i is a *k*+*v* length vector having 1 at i^{th} location and 0 everywhere else. For i=1, a keystream is generated using this new key/IV pair and XORed with the initial keystream. The result is stored as a row of a matrix. The above process is repeated for *N* different key/IV pair and result is stored as a row of same matrix. In this way, a matrix of order $N \times L$ is produced. Then the hamming weight of each row is calculated. These *N* hamming weights should follow Binomial distribution with (*L*, 1/2). These *N* weights are grouped into 5 different categories and Chi-Square test is performed. For each e_i, a matrix is generated using same procedure and Chi-Square test is performed to analyze the data.

To perform this test, 1024 different key/IV pair are chosen randomly and keystream of 1024-bit is generated for each of these key/IV pair. The above process is followed and a matrix of order 1024 × 1024 is generated. The row weight is calculated and grouped into 5 categories. The Chi-Square test is applied to get the *p*-value. Similarly, $(k+v)$ -1 more *p*-values are calculated using e_i where $i = 2, \ldots, (k+v)$ and if *p*-value is less than 0.01 for i^{th} matrix, it means that i^{th} bit of the key/IV pair is not properly mixed. Table 9, 10 and 11 shows the matrix whose *p*-value is less than 0.01 and its corresponding *p*-value obtained.

Table 9. ZUC: SAC-r diffusion test

Matrix No.	*p*-Value
51 (51st bit of key)	0.0071
77 (77th bit of key)	0.0041
187 (59th bit of IV)	6.9901×10^{-05}

Table 10. Espresso: SAC-r Diffusion Test

Matrix No.	*p*-Value
15 (15th bit of key)	0.0060
81 (81st bit of key)	0.0039
133 (5th bit of IV)	0.0070
165 (37th bit of IV)	0.0062
200 (72nd bit of IV)	0.0020

Table 11. Grain v1: SAC-r diffusion test

Matrix No.	*p*-Value
9 (9th bit of key)	0.0036

G. SAC-c DIFFUSION TEST

In SAC-c, similar to SAC-r, a matrix is obtained by XORing key/IV pair with e_i and generating L bit of keystream which is XORed with original keystream before storing it in a row of matrix. Again, each matrix is of order $N\times L$. But, in SAC-c, the hamming weight of each column is calculated instead of row, as in case of SAC-r. These L hamming wieghts are then groupd into 5 categories and χ^2 test is applied to analyze the result.

Table 12 and 13 shows the matrix of ZUC and Espresso whose *p*-values are less than 0.01. We didn't find any such matrix in Grain v1.

Table 12. ZUC: SAC-c diffusion test

Matrix No.	*p*-Value
61 (61st bit of key)	0.0093
187 (59th bit of IV)	0.0004
251 (123rd bit of IV)	0.0070
254 (126th bit of IV)	0.0070

Table 13. Espresso: SAC-c diffusion test

Matrix No.	*p*-Value
200 (72nd bit of IV)	0.0041

H. k-ERROR LINEAR COMPLEXITY

Linear complexity is useful in determining the randomness of pseudo-random sequences. Linear complexity of a pseudo-random sequence is defined as the number of stages required in shortest LFSR to generate the sequence. It should be taken care that cryptographically strong sequences should have high linear complexity.

High linear complexity is necessary but not sufficient condition for a sequence to be cryptographically strong [12]. Therefore, additional tests are required to determine the strength of a pseudo-random sequence.

Mark Stamp and Clyde F. Martin proposed a method named "*k*-error linear complexity" that can help to determine the strength of pseudo-random sequences [9]. The *k*-error linear complexity is the linear complexity of a sequence when *k* or fewer errors occur in the sequence. So, 0-error linear complexity is just the linear complexity of the sequence [14].

The *k*-error linear complexity profile is plotted with *k*-error linear complexity against k value. This profile is a step function which can be used in distinguishing keystreams generated using different cryptographic algorithms.

For *k*-error linear complexity, 1024 bit keystream of all three ciphers are generated and complexity is obtained by inducing errors. Fig. 4 shows the *k*-error complexity profile of all three ciphers.

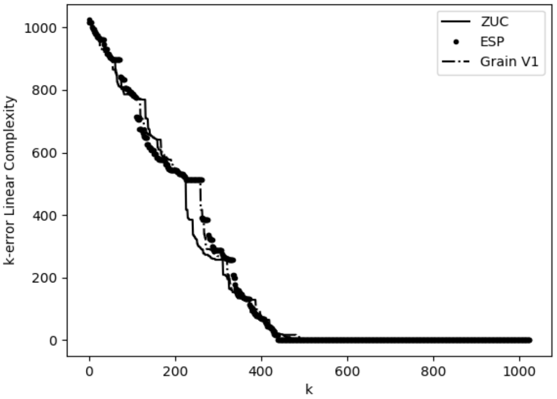

Figure 4. *k*-error linear complexity

Along with this, steps are calculated for all ciphers for 1024 different key/IV pair and average has been shown in Table 14.

Table 14. Average steps of *k*-error linear complexity

Ciphers	Steps (Average)
ZUC	512.8730
Espresso	511.9629
Grain v1	512.1504

I. NON-LINEAR COMPLEXITY

The non-linear complexity, also called maximum order complexity of a pseudo random sequence, gives the length of shortest NLFSR required to generate it.

The non-linear complexity can be obtained by finding the length of the longest sub-sequence L which is repeating at least twice and have different successor [8]. Then its non-linear complexity is defined as $L+1$. The ideal non-linear complexity for n-bit random sequence is $2 \log_2 n$.

For non-linear complexity, Keystreams of length 2^{10} are generated using random Key/IV pair and non-linear complexity is calculated for 10^3 different keystreams. Average of non-linear complexity of each cipher is shown in Table 15.

Table 15. Non-linear complexity

Ciphers	Non-linear Complexity
ZUC	19.2344
Espresso	19.2256
Grain v1	19.1934

IV. PROPOSED TESTS AND RESULTS

All the statistical tests described above are based on choice of key/IV having random hamming weights. To check the effect of low/high hamming weight of key and IV on internal state, we have proposed four new tests in this paper.

A. INTERNAL STATE CORRELATION USING LOW HAMMING WEIGHT KEY

The purpose of this test is to analyze the effect of key with low hamming weight on the internal state of the stream ciphers. For this test, 2^{20} different keys with 30% hamming weight were generated and internal states were obtained keeping the IV constant. Each internal state is stored as row of a matrix, thus the matrix created is of order 2^{20} × (size of internal state). The internal state of ZUC, Espresso and Grain v1 are of 496, 256, 160-bits respectively. The row weight of the matrix is calculated and categorized in 5 groups. The weight should follow Binomial distribution (n, p) where n= size of internal state and p = 1/2. The groups and their probabilities are shown in Table 16 and 17.

Table 16 – Weight category and probability for 496 bit and 256 bit

For 496-bits		For 256-bits	
Category Limit	**Probability**	**Category Limit**	**Probability**
0-238	0.1968	0-121	0.2083
239-245	0.2144	122-126	0.2174
246-251	0.2121	127-130	0.1970
252-258	0.2038	131-135	0.2031
259-496	0.1729	136-256	0.1743

Table 17 – Weight category and probability for 160 bit

For 160-bits	
Category Limit	**Probability**
0-74	0.1923
75-78	0.2140
79-82	0.2473
83-86	0.1944
87-160	0.1520

Chi-Square test is applied to evaluate the correlation between internal states.

As shown in Table 18, all three ciphers have p-value greater than 0.01 which means that low key weight doesn’t induce any weakness in the internal state.

Table 18. Internal state correlation using low hamming weight key

Ciphers	*p*-value
ZUC	0.8796
Espresso	0.1799
Grain v1	0.7074

B. INTERNAL STATE CORRELATION USING LOW HAMMING WEIGHT IV

This test is similar to *internal state correlation using low hamming weight key*. But instead of key, here, 2^{20} different IVs with 30% hamming weight were generated and internal states were obtained keeping the key constant. Same process is followed as mentioned in above test and the Chi-Square test is used to find the correlation between internal states.

As shown in Table 19, all three ciphers have p-value greater than 0.01 which means that low IV weight doesn’t induce any weakness in the internal state.

Table 19. Internal state correlation using low hamming weight IV

Ciphers	*p*-value
ZUC	0.8746
Espresso	0.3886
Grain v1	0.6912

C. INTERNAL STATE CORRELATION USING HIGH HAMMING WEIGHT KEY

The purpose of this test is to analyze the effect of key with high hamming weight on the internal state of the stream ciphers. For this test, 2^{20} different keys with more than 80% hamming weight were generated and internal states were obtained keeping the IV constant. Each internal state is stored as row of a matrix, thus the matrix created is of order 2^{20} × (size of internal state). The row weight of the matrix is calculated and categorized in 5 groups. The groups and their probabilities are shown in Table 16 and 17. The Chi-Square test is applied to evaluate the correlation between internal states.

As shown in Table 20, all three ciphers have p-value greater than 0.01 which means that high key weight doesn’t induce weakness in the internal state.

Table 20. Internal state correlation using high hamming weight key

Ciphers	*p*-value
ZUC	0.4375
Espresso	0.8267
Grain v1	0.8519

D. INTERNAL STATE CORRELATION USING HIGH HAMMING WEIGHT IV

In this test, 2^{20} different IVs with more than 80% hamming weight were generated and internal states were obtained keeping the key constant. The same process is repeated as in above test and the Chi-Square test is applied to evaluate the correlation between internal states.

Table 21 shows the p-values for all three ciphers. Grain v1 has p-value less than 0.01, which means that we need to revise the IV loading part of Grain v1 stream cipher such that high weight IVs are not loaded.

Table 21. Internal state correlation using high hamming weight IV

Ciphers	*p*-value
ZUC	0.2537
Espresso	0.4718
Grain v1	6.0951×10^{-07}

V. CONCLUSION

All the three ciphers failed to pass the diffusion test. To analyze more about this failure, SAC-r and SAC-c diffusion tests were performed. Here we came to know that flipping some particular bits of key or IV don't have expected influence on the keystream. Table 9 shows that flipping the 51st and 77th bit of key and flipping the 59th bit of IV do not influence the keystream for ZUC. Similarly for Espresso, bit flip in key at 15th and 81st bit and bit flip in IV at 5th, 37th and 72nd bit are not influencing the keystream. For Grain v1, only 9th bit of key is not influencing the keystream. SAC-c diffusion test was performed using the same key/IV pair which was used in SAC-r test. As shown in Table 12 and 13, some of the bit flips of key and IV of ZUC are failing the Chi-Square test and flip at 72nd bit of IV of Espresso is also failing this test.

In this paper, we have proposed four different tests to check the correlation of internal states using key and IV with high or low hamming weight. All three ciphers passed three of the tests but Grain v1 cipher failed the 4th test. This shows that if hamming weight of IV of Grain v1 is very high, the produced keystream is deviating from true randomness.

References

[1] N. Yerukala, V. Kamakshi Prasad, and A. Apparao, "Performance and statistical analysis of stream ciphers in GSM communications," *Journal of Communications Software and Systems*, vol. 16, issue 1, pp. 11-18, 2020.

[2] E. Dubrova, and M. Hell, "Espresso: A stream cipher for 5G wireless communication systems," *Cryptography and Communications*, vol. 9, issue 2, pp. 273-289, 2017.

[3] M. Hell, T. Johansson, and W. Meier, "Grain: a stream cipher for constrained environments," *International Journal of Wireless and Mobile Computing*, vol. 2, issue 1, pp. 86-93, 2007.

[4] M. U. Bokhari, S. Alam, and S. H. Hasan, "A detailed analysis of Grain family of stream ciphers," *Int J Comput Netw Inf Secur*, vol. 6, pp. 34-40, 2014.

[5] Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 128-EIA3. Document 2: ZUC Specification.

[6] M. S. Turan, A. Doğanaksoy, and Ç. Çalik, "Statistical analysis of synchronous stream ciphers,", Proceedings of the International Conference SASC 2006: Stream Ciphers Revisited, 2006, pp. 84-93.

[7] C. Srinivassan, K. V. Lakshmy, and M. Sethumadhavan, "Measuring diffusion in stream ciphers using statistical testing methods," *Defence Science Journal*, vol. 62, issue 1, 6, 2012.

[8] K. Limniotis, N. Kolokotronis, and N. Kalouptsidis, "On the nonlinear complexity and Lempel–Ziv complexity of finite length sequences," *IEEE Transactions on Information Theory*, vol. 53, issue 11, pp. 4293-4302, 2007.

[9] M. Stamp, and C. F. Martin, "An algorithm for the k-error linear complexity of binary sequences with period 2/sup n," *IEEE Transactions on Information Theory*, vol. 39, issue 4, pp. 1398-1401, 1993.

[10] S. Lakshmi, et al., "A quasigroup based synchronous stream cipher for lightweight applications," *Proceedings of the International Symposium on Security in Computing and Communication*, Springer, Singapore, 2017, pp. 205-214.

[11] M. Abumuala, O. Khalifa, and A.-H. A. Hashim, "A new method for generating cryptographically strong sequences of pseudo random bits for stream cipher," *Proceedings of the IEEE International Conference on Computer and Communication Engineering (ICCCE'10)*, 2010, pp. 1-4.

[12] F. Zhu, and W. Qi, "Thek-error linear complexity and the linear complexity forpq n-periodic binary sequences," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1549-1553, 2006.

[13] C. Zhou, X. Feng, and D. Lin, "The initialization stage analysis of ZUC v1. 5.", *Proceedings of the International Conference on Cryptology and Network Security*, Springer, Berlin, Heidelberg, 2011, pp. 40-53.

[14] A. G. B. Lauder, and K. G. Paterson, "Computing the error linear complexity spectrum of a binary sequence of period 2n," *IEEE Transactions on Information Theory*, vol. 49, issue 1, pp. 273-280, 2003.

[15] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, issue 3, pp. 1617-1655, 2016.

[16] https://github.com/gulshanRaj/Grain_V1_impl-ementation

[17] https://github.com/lemi101/Espresso_Impleme-ntation

[18] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, issue 3, pp. 1617-1655, 2016.

[19] C.-Y. Li, et al., "Insecurity of voice solution volte in lte mobile networks," *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 316-327.

[20] ETSI/SAGE Specification. Specification of the GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3.Document 1: 128-EEA3 and 128-EIA3 Specification, 2011.

[21] M. Hell, et al., "A stream cipher, proposal: Grain-128," *Proceedings of the 2006 IEEE International Symposium on Information Theory*, 2006, pp. 1614-1618.

Mr. ***SAURABH SHRIVASTAVA*** *is currently pursuing his M. Tech in Cyber Security from Amrita Vishwa Vidyapeetham University Coimbatore. His area of research is Cryptography.*

Dr. ***K. V. LAKSHMY*** *has received her Ph.D. in Mathematics from Amrita Vishwa Vidyapeetham, Coimbatore. Currently she is working as an Assistant Professor at TIFAC-CORE in Cyber Security, Coimbatore. Her area of research is cryptanalysis.*

Dr. ***CHUNGATH SRINIVASAN*** *has obtained his Ph.D. in Mathematics from Amrita Vishwa Vidyapeetham, Coimbatore. Currently he is working as an Assistant Professor at TIFAC-CORE in Cyber Security, Coimbatore. His area of research are Cryptology, Coding and Information Theory.*