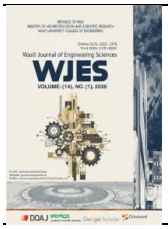

Wasit Journal of Engineering Sciences
Journal homepage: https://ejuow.uowasit.edu.iq

ISSN 2305-6932
Original Research

Vol. 14 No. 1 March 2026, pp.25-36
https://doi.org/10.31185/ejuow.v14i1.762

A Novel Method in Hybrid Encryption for Enhanced 5G Networks Security

Ali Ahmed Mustafa[1], Lujain Sabah Abdulla[1]

[1] Department of Electrical Engineering, College of Engineering, Tikrit University, Tikrit, Iraq

Corresponding Author Email: aa230050en@st.tu.edu.iq

Received Jul.17, 2025 | Revised Aug.8, 2025 | Accepted Sep.3, 2025 | Online March.1, 2026

ABSTRACT

Encryption is one of the most important security methods in Fifth Generation (5G) networks, offering high speed and security to support enhanced Mobile Broadband (eMBB), as well as Ultra-Reliable Low-Latency Communications (URLLC). Each algorithm (Advanced Encryption Standard (AES), Data Encryption Standard (DES), ZUC, Synchronous Network-Oriented Word-oriented stream cipher, version 3G (Snow3G)) has distinct strengths and weaknesses in speed and security. Therefore, this research presents two new hybrid models: ZUC-AES (combining the speed of ZUC with the strength of AES) and SNOW3G-DES (combining the speed of SNOW3G with the strength of DES). The models were simulated in MATLAB for data up to 5,000,000 bits, the results simulating six algorithms in MATLAB showed an improvement in strength compared to conventional algorithms, with the p-value on the National Institute of Standards and Technology (NIST) SP 800-22 test improving by approximately average 50% for ZUC-AES (0.4323) compared to ZUC alone (0.288). ZUC was found to be the fastest of all with time 0.85s to 1000000 bit, while the third model, ZUC-AES, achieved superior results compared to AES and DES, while closely matching Snow3G's performance. The study proposed integrating RSA into the system to encrypt keys for secure exchange and increase the strength of encryption

Keywords: Encryption, Five generation (5G), ZUC, NIST

1. Introduction (11 PT)

5G networks increasingly rely on high-speed, low-latency communications, making the security of data transmission vital to ensure confidentiality and integrity [1-3]. Traditional encryption algorithms such as AES and DES remain prominent methods for protecting data: AES offers strong security and high software efficiency [4,5], while DES's short key length leaves it vulnerable to brute-force attacks [6,7]. Stream algorithms like SNOW3G and ZUC, defined in 3rd Generation Partnership Project (3GPP) specifications, meet encryption requirements in 4G and 5G environments: ZUC excels in speed and mobile efficiency, whereas snow3G provides enhanced resilience [8,9]. Despite these advances, no existing method achieves the optimal balance between low encryption latency and robust security required for URLLC and eMBB services in 5G networks. Symmetric ciphers (AES, DES) are fast but susceptible to advanced analytical attacks, while asymmetric schemes (RSA, Elliptic Curve Cryptography (ECC)) deliver strong security at the cost of high processing times incompatible with 5G performance demands. Therefore, this research proposes two hybrid stream-block encryption models that leverage the security strengths of AES and DES alongside the speed advantages of ZUC and SNOW3G. The first model integrates AES's cryptographic robustness with ZUC's lightweight keystream generation, and the second model combines SNOW3G's rapid stream cipher operations with a single round of DES for enhanced diffusion. We evaluated both hybrids against AES, DES, SNOW3G, and ZUC in terms of encryption/decryption time, and keystream randomness. Results demonstrate that the proposed models achieve a superior security-speed trade-off across successive encryption rounds. These findings indicate the potential to enhance 5G network security without sacrificing performance, thereby supporting high-throughput IoT deployments and other low-latency applications.

In 2021, Yang, Jing [10] demonstrated that SNOW-3G with a 256-bit key was vulnerable (with attacks of complexity (2^{172} and 2^{177}) thus developed SNOW-V, achieving 256-bit security with speeds up to 58 Gbps. In 2020, Rabia Khan et al [11] surveyed core technologies for a 5G security model, discussing network monitoring, management, standards, key projects, and future research directions. Later in 2021, Ardalan H. Awlla and Sirwan M. Aziz [12] enhanced AES with shift row and Mix Columns, outperforming the traditional version.
This paper is organized as follows: Section 2 details our proposed approach, including the system model, and implementation architecture. Section 3 describes the results and critical discussion. Section 4 concludes the paper and suggests directions for future research.

1. System model

In this section, six commonly used encryption algorithms were examined to identify their strengths and weaknesses, assess security and performance, and determine their strength and speed to address existing problems with the algorithms. They were simulated in MATLAB. After obtaining the results for the six algorithms, two hybrid ciphers were created to leverage their advantages. The strength of AES and the speed of ZUC were combined in one hybrid cipher, and the strength of DES and the speed of Snow3G were combined in another hybrid cipher. The six algorithms studied were (AES, DES, ZUC, snow3G), and the two ciphers created were (ZUC-AES, snow3G-DES). The results will be presented later in section 3.

1.1. Research methodology

We discussed the methodology of the algorithms AES, DES, ZUC, Snow3G and the proposed designs.

2.1.1 Advanced encryption standard (AES)

It is a symmetric algorithm with a 128-bit block size and three different key sizes: 128, 192, and 256, with rounds of 10, 12, and 14, respectively [13] Fig. 1 shows AES Algorithm.

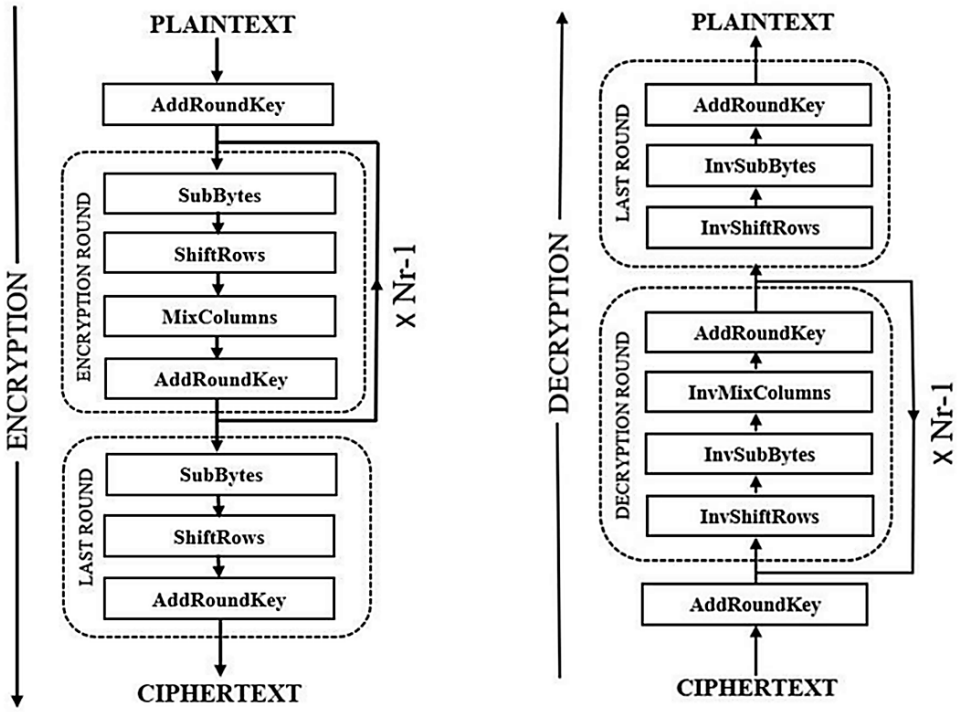

Figure 1. AES Algorithm [14]

The block is stored in a 4*4 matrix called state. After that, an XOR is performed with the key. After that, several rounds of the following operations are performed:

1. SubByte replace the values resulting from the XOR operation with values in table
2. Shift-Rows by 0, 1, 2, and 3 positions to the left
3. Mixcolums
4. AddRoundKey This operation applies the bitwise addition of the round key $\boldsymbol{k} \in \mathbb{F}_2^{128}$ to the state $\boldsymbol{x} \in \mathbb{F}_2^{128}$, i.e., AddRoundKey: $\boldsymbol{x7} \rightarrow \boldsymbol{x} \oplus \boldsymbol{k}$. Except for the last round, which is performed without Mixcolums.

The number of rounds depends on the key size, as mentioned above. As for the receiving end, during decryption, the same process happens but in reverse.

2.1.2 Data Encryption Standard (DES)

A 64-bit block algorithm with a 64-bit key, 56 bits of which are used for encryption and 8 bits for parity check. With 16 rounds as shown in Fig. 2, the algorithm begins by passing the (64)-bit block to the Initial Permutation (IP), where 64 bits remain, but with different values [15]. After the rounds are completed, the final data is passed to the Final Permutation (FP). After the IP is completed, the result is divided into two parts: L0(33-64) and R0(1-32) for 16 rounds of operations, including

$$L_i = R_{i-1} \quad (1)$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \quad (2)$$

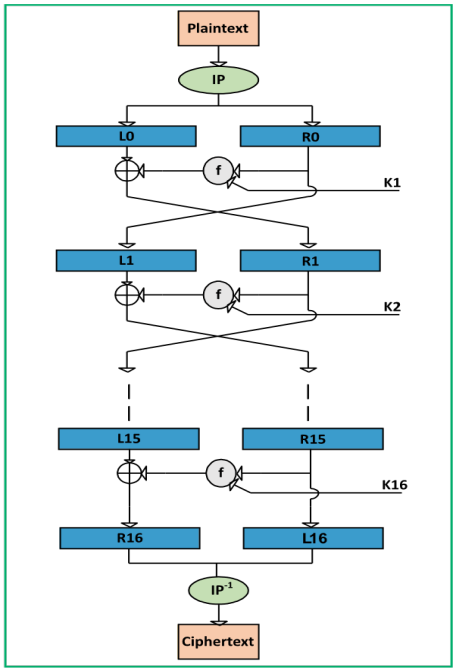

Figure 2. DES Algorithm [16]

2.1.3 PC-1 Permuted Choice 1

Remove 8 bits from the original key (64 bits) to create a 56-bit subkey [17] and then Split into two halves (C_0, D_0): each of 28 bits.

$$(C_0 \| D_0) = PC-1(Key^{64-}bit) \rightarrow 56 \quad (3)$$

$$K_i = PC-2(C_i \| D_i) \rightarrow 48 \quad (4)$$

Ciphertext is final permutation to $(L_{16} \| R_{16})$.

In Decryption end using the same steps as above, but with the sub keys in reverse order ($K_{16}$to K_1). All equations and figures remain the same, with no changes other than the order of the keys

2.1.4 Zuc-128

The algorithm consists of three layers: The Linear Feedback Shift Register (LFSR) register, the second is the Reorganization (BR), the third is the nonlinear function F [18], linear feedback shift register: The first layer performs the following operations:
Distributing the 128-bit key and the 128-bit vector iv and the constant D

$$k = k_0||k_1||..||k_{15} \quad (5)$$

and

$$iv = iv_0||\, iv_1||\, iv_2\, ||\, ...\, ||\, iv_{15} \quad (6)$$

$$D = d_0||d_1\, ||\, ...\, ||d_{15}, \quad (7)$$

$$s_i = k_i||d_i||iv_i \quad (8)$$

with 15 bits per register
Let D be a 240-bit long constant string composed of 16 substrings of 15 bits:
Over 16 registers as shown on Fig. 3. each with a size of 31 bits, for a total of 496 bits with the following order:

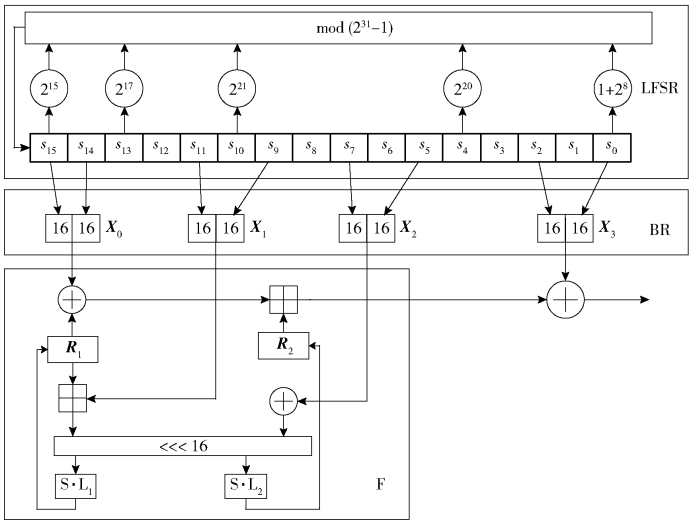

Figure 3. ZUC-128 Encryption and Decryption Process

The LFSR has 2 modes: the initialization mode and the working mode. In the initialization mode, the LFSR receives a 31-bit input word u, which is obtained by removing the rightmost bit from the 32-bit output W of the nonlinear function F, i.e. More specifically, the initialization mode works as follows: LFSR with Initialization Mode(u)

$$V = 2^{15}S_{15} + 2^{17}S_{13} + 2^{21}S_{10} + 2^{20}S_4 + (1 + 2^8)S_0\ mod\ (2^{31} - 1) \quad (9)$$

and (10)

$$S_{16} = (v + u)\ mod\ (2^{31} - 1); \quad (10)$$

If S_{16}=0, then set S_{16}=$2^{31} - 1$

$$(S_1, S_2, \dots, S_{15}, S_{16}) \rightarrow\ (S_0, S_1, \dots, S_{14}, S_{15}). \quad (11)$$

In the working mode, the LFSR does not receive any input, and it works as follows: LFSR with work mode

$$S_{16} = 2^{15}S_{15} + 2^{17}S_{17} + 2^{21}S_{10} + 2^{20}S_4 + (1 + 2^8)S_0\ mod\ (2^{31} - 1) \quad (12)$$

$$(S_1, S_2, \dots, S_{15}, S_{16}) \rightarrow\ (S_0, S_1, \dots, S_{14}, S_{15}). \quad (13)$$

The bit-reorganization
The middle layer of the algorithm is the bit-reorganization. It extracts 128 bits from the cells of the LFSR and forms 4 of 32-bit words, where the first three words will be used by the nonlinear function F in the bottom layer, and the last word will be involved in producing the keystream.
Let S_0, S_2, S_5, S_7, S_9, S_{11}, S_{14}, S_{15} be 8 cells of LFSR Then the bit reorganization forms 4 of 32-bit words X0, X_1, X_2, X_3 from the above cells as follows:

$$\begin{aligned} X_0 &= S_{15H}||S_{14L} \\ X_1 &= S_{11L}||S_{9H} \\ X_2 &= S_{7L}||S_{5H} \\ X_3 &= S_{2L}||S_{0H}. \end{aligned} \tag{14}$$

The nonlinear function F utilizes two 32-bit memory cells (R_1 and R_2). Let the inputs to F be X_0, X_1 and X_2, which come from the outputs of the bit-reorganization, then the function F outputs a 32-bit word W. The detailed process of F is as follows: $F\ (X_0, X_1, X_2)$

$$W = (X_0 \oplus R_1) \boxplus R_2 \tag{15}$$

where $\boxplus$ main $mod\ (2^{32})$

$$W_1 = R_1 \boxplus X_1 \tag{16}$$

$$W_2 = R_2 \oplus X_2 \tag{17}$$

$$R_1 = S(L_1(W_{1L}||W_{2H})) \tag{18}$$

$$R_2 = S(L_2(W_{2L}||W_{2H})) \tag{19}$$

where S is a 32×32 S-box, L_1 and L_2 are linear transforms The S-box S The 32×32 S-box S is composed of 4 juxtaposed 8×8 S-boxes, i.e., S to (S_0, S_1, S_2, S_3), where S_0=S_2, S_1=S_3. Let x be an 8-bit input to S_0 (or S_1). Write x into two hexadecimal digits, then the entry at the intersection of the h-th row and the l-th column in table (or table) is the output of S_0 (or S_1).
The linear transforms L_1 and L_2: Both L_1 and L_2 are linear transforms from 32-bitwords to 32-bit words, and are defined as follows:

$$L_1(X) = X \oplus (X <<<_{32} 22) \oplus (X <<<_{32} 10) \oplus (X <<<_{32} 18) \oplus (X <<<_{32} 24) \tag{20}$$

$$L_2(X) = X \oplus (X <<<_{32} 8) \oplus (X <<<_{32} 14) \oplus (X <<<_{32} 22) \oplus (X <<<_{32} 30) \tag{21}$$

After 32 cycles without production (initialization cycles), the stream key is produced in the following form, consisting of 32 bits [18].

$$Z = W \oplus X_3 \tag{22}$$

2.1.5 3.5 Snow 3G

A streaming algorithm like ZUC, except that it has three internal R registers and only two layers: LFSR and Finite State Machine (FSM). The key and vector are first partitioned across 16 registers (each register is 32 bits long) Then Calculate a new value for S16 using the following equation:

$$s_{t+16} = (\alpha \cdot S_t) \oplus s_{t+2} \oplus (\alpha^{-1} \cdot s_{t+11}) \tag{23}$$

α = 0x00000002 and α^{-1} such that

$$\alpha \times \alpha^{-1} \equiv 1\ (mod\ P(x)) \tag{24}$$

Then rotate the digits to the right. Shift: $s_i \leftarrow s_{i+1}$ (i=0…14), $S_{15} \leftarrow S_{16}$. The FSM includes the following operations with update in every round [19]:

$$X_1 = R_3 \oplus S_5 \tag{25}$$

$$R_1 = (X_1 \boxplus R_2) \tag{26}$$

$$R_2 = S\_box1\ (R_1) \tag{27}$$

$$R_3 = S\_box2\ (R_2) \tag{28}$$

$$W = (S_{15} \boxplus R_1) \oplus R_2 \tag{29}$$

After 32 cycles without production the stream key is produced in the following form and as shown in Fig. 4 Keystream word:

$$Z = W \oplus S_0 \tag{30}$$

Throughout key generation, the SNOW 3G algorithms are used Fig. 4

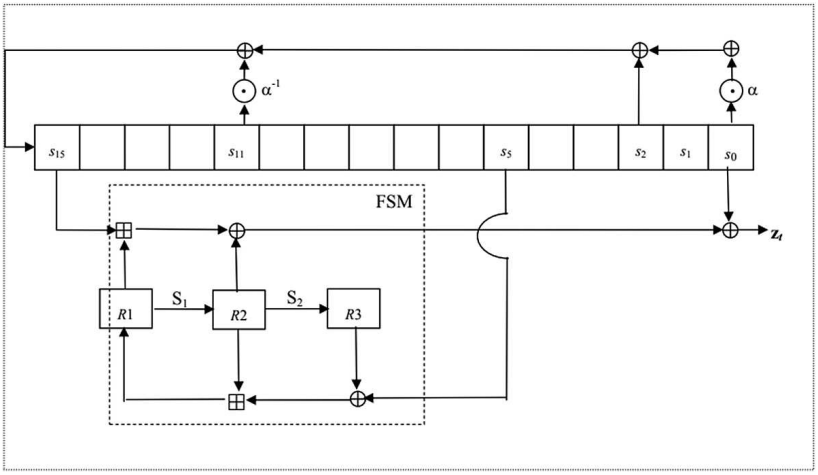

Figure 4. Snow3G Algorithm Throughout Key Generation [20]

1.2. Proposed designs

1.2.1. ZUC-AES

Figure 5 explain This hybrid algorithm, is a new and previously unexplored part, combining two algorithms: one for block and the other for stream. This leverages the power of AES. The speed of ZUC is well-known for stream algorithms. Stream algorithms. ZUC operates alone during the 32 initialization cycles. AES is inserted between the BR layer and the F layer, where the outputs of (X_0, X_1, X_2, X_3) from the BR layer, consisting of 128 bits (4*32), are taken. This provides a suitable input for Aes, which encrypts these inputs at each stream key generation cycle, producing 128 encrypted bits. This is then divided into four groups of 32 bits each, replacing the previous group (X_0, X_1, X_2, X_3) in order.

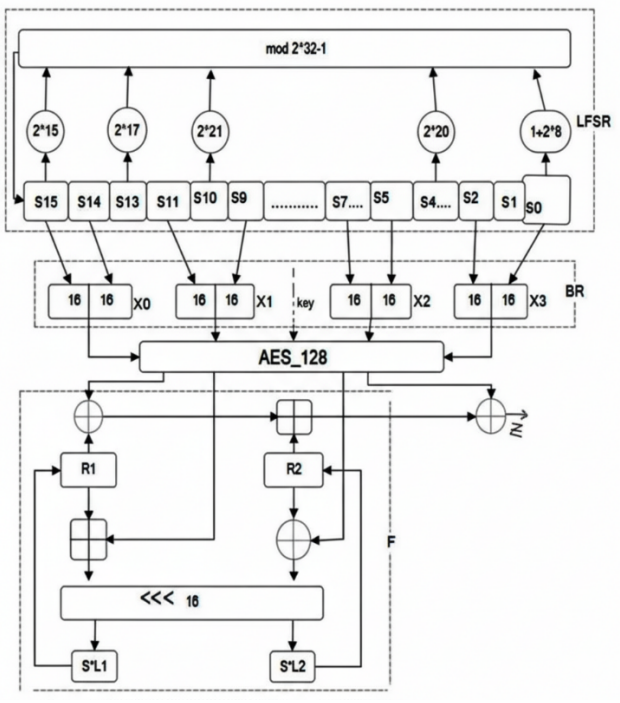

Figure 5. ZUC_ AES Hybrid Encryption and Decryption Process

1.2.2. Snow3G_DES

A hybrid cipher combining DES's security strength with Snow3G's high speed. The algorithm operates as snow3G alone during initialization cycles. During key stream generation cycles, des is introduced into the snow3G algorithm by passing S_{15} and S_5 (2*32 bit) as a 64-bit block that is encrypted during each key stream generation cycle as shown in Fig. 6. This also produces a 64-bit encrypted block that is divided into two parts:

The right part goes where S_5 was going, and the left part goes where S_{15} was going. We previously discussed the two algorithms separately. When the algorithm was executed, the following result were obtained.

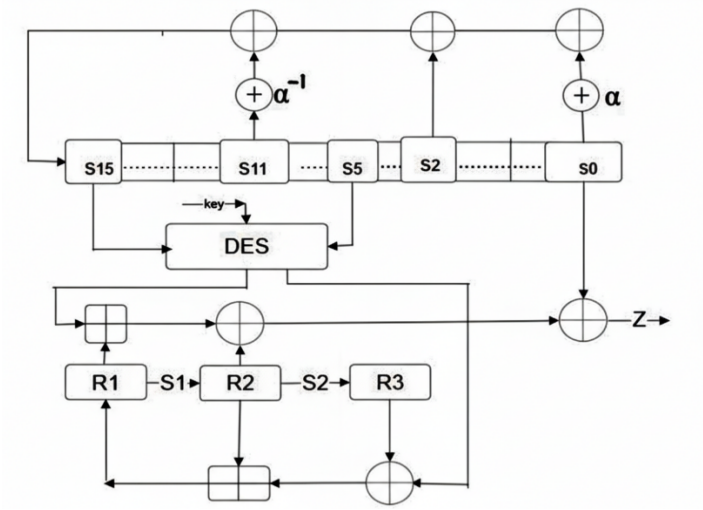

Figure 6. Snow3G_DES Hybrid Encryption and Decryption Process

3. Simulation results

In this section of the research, the results of simulating six algorithms in MATLAB were presented, four of which were single and two proposed hybrids, for the purpose of measuring the encryption and decryption time in each of them in seconds. In the streaming algorithms that produce a streaming key, only one reading is adopted, which is the encryption time, because the decryption time is considered the same because it depends on the same steps. The results were as follows:

3.1. Non-hybrid encryption algorithms

When simulating AES, DES, ZUC and Snow3G algorithms we obtained the results in Table 1.

Table 1. AES, DES, ZUC, Snow3G Encryption and Decryption Time

Data Size(bit)	AES Enc time	AES Dec time	DES Enc time	DES Dec time	ZUC Enc time	Snow3G Enc time
1000	0.0039	0.0036	0.0077	0.0096	0.003	0.007
64000	0.21	0.18	0.36	0.33	0.067	0.17
128000	0.38	0.34	0.65	0.63	0.126	0.3
256000	0.71	0.67	1.28	1.26	0.25	0.6
512000	1.37	1.34	2.5	2.51	0.46	1.15
1000000	2.68	2.67	4.97	4.86	0.85	2.06

The encryption time results for AES across six data sizes show a near-linear relationship between bit size and encryption/decryption time. For instance, with 256,000 bits, encryption took 1.28s and decryption 1.26s. Doubling the data to 512,000 bits nearly doubled the time. Among the symmetric algorithms tested, AES is the fastest, yet it is slower than the stream ciphers ZUC and Snow3G, while DES is the slowest. When simulating DES in MATLAB with optimized function calls, we observed similar near-linear scaling. At 256,000 bits, DES took 1.28s and 1.26s; at 512,000 bits, times increased to 2.5s and 2.51s respectively. However, results for small inputs (e.g., 1,000 or 64,000 bits) were inconsistent due to measurement variability.

Table 1 also includes ZUC encryption results. Only encryption time is shown, as it equals decryption time due to identical operations (stream key generation and XOR). ZUC had the shortest encryption time among all algorithms. For example, at 128,000 bits it took 0.126s, and at 256,000 bits, 0.25s—again showing near-linear growth. Snow3G follows a similar pattern, though slower than ZUC. It required 2.06s to encrypt 1,000,000 bits,

compared to 0.85s for ZUC, but still faster than AES (2.68s) and DES (4.97s). Like ZUC, Snow3G's encryption time increases proportionally with data size.

3.2. Proposed hybrid encryption algorithms

In ZUC_AES Three models of ZUC_AES encryption were created. In addition to the original model (the first), two other models were extracted from it. The first is to make the AES call limited to only one round instead of 10 rounds. The second is that in addition to calling one AES round, AES is called once out of four times to produce a stream key. That is, the hybrid algorithm works as ZUC alone to produce three stream keys and calls one round of AES when producing the fourth stream key. The calling is done alternately and is not limited to a specific order. Also, four subkeys are used, that is, a subkey for each call within 128 bits. The same thing was done with snow3G_DES. In addition to the original design (the first), two other models were extracted from it. The first is to make the des algorithm limited to one round instead of 16, and the second is to make the hybrid design work once as snow3G only and once des is called for only one round. The results for the three models for both ZUC_AES and snow3G_DES are as in Table 2.

Table 2. The Results for the Three Models for Both ZUC_AES and snow3G_DES

Hybrid	Data size (bit)	First design enc time (second)	Second design enc time (second)	Third design enc time (second)
ZUC_AES	1000	0.021	0.024	0.022
	64000	0.43	0.36	0.2
	128000	0.78	0.61	0.37
	256000	1.4	1.05	0.7
	512000	2.72	1.95	1.2
	1000000	5.2	3.55	2.1
	5000000	28	16.63	9.61
Snow3G_DES	1000	0.021	0.013	0.0085
	64000	0.74	0.24	0.19
	128000	1.47	0.47	0.36
	256000	2.9	0.91	0.68
	512000	5.83	1.81	1.35
	1000000	11.32	3.52	2.61
	5000000	63.6	17.13	13

The results of the hybrid encryption (First design) simulation for the first proposed design by introducing the AES algorithm into ZUC, show that the encryption time of the hybrid algorithm ZUC-AES is approximately 1.5 times, or one and a half times the encryption time of the two algorithms combined. For example, the encryption time for 512,000 bits in AES was 1.37 seconds, in ZUC it was 0.46 seconds, and in the first proposed design it was 2.72 seconds. When dividing this time by the encryption times of AES and ZUC (1.37 + 0.46 = 1.83), the result is 2.72 ÷1.83 = 1.48. It is true that AES and ZUC are called 4 times for each call for the same data size if it was done with the AES algorithm, because the size of the data encryption block is 128 bits, while the flow key for one cycle is 32 bits, meaning four call cycles in the proposed encryption compared to one cycle in AES alone. Despite all this, as we said, it is equivalent to 1.5 times for them, and the reason for this is that the time taken in the first cycle is not the same as the time for the cycle 1000 for example because the two algorithms are initialized, for example, matrices, as well as s-box and the rest of the operations such as XOR, and thus it becomes faster. The same thing mentioned above in the first proposed design also applies to the second proposed algorithm (Snow3g_DES first design), which is inserting des into snow3G. Despite invoking DES and Snow3G twice per 64-bit block (vs. once in standalone DES), encryption time did not double and once in the case of the des algorithm alone, the encryption time did not double for the same reasons mentioned in the ZUC_AES results.

3.3. Comparisons between proposed algorithms with traditional algorithms

The Fig. 7 below shows Proposed ZUC-AES algorithms with other algorithms encryption time.

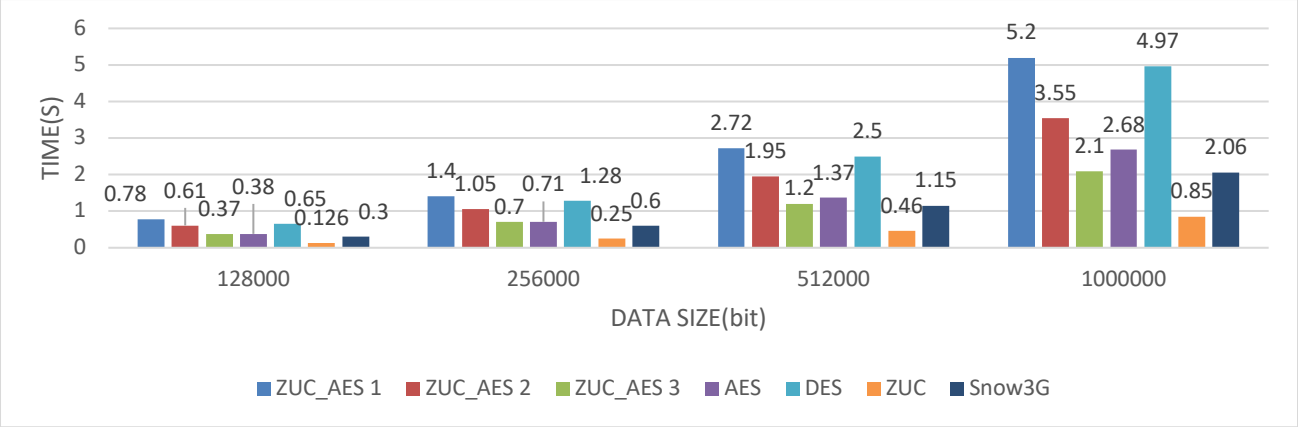

Figure 7. Three Proposed Designs for ZUC-AES with other Algorithms

The Fig. 8 below shows the three proposed designs for Snow3G_DES with other algorithms.
It is clear from the results in above that the third model of the proposed design ZUC_AES outperforms AES alone in speed and is very close to the speed of snow3G, but ZUC is still the fastest among all, while the first design is the slowest among all, but the strongest (2^{384}) compare with 2^{256} in ZUC [18] among them. The same applies to snow3G -DES, as the third design also outperforms AES in speed, and the second design is faster than DES alone, while the first design is slower but stronger than the single one (2^{320}).

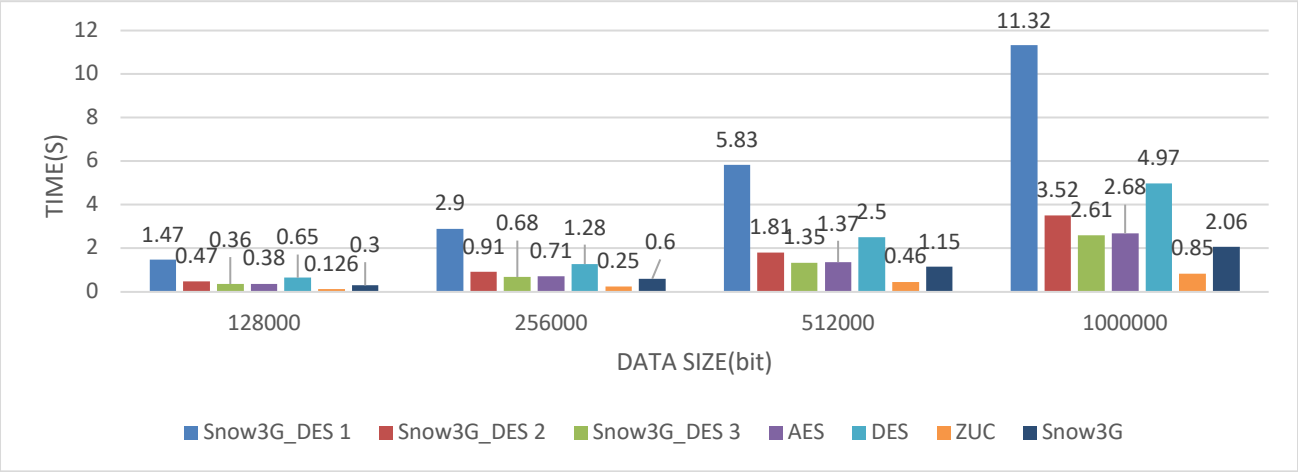

Figure 8. Three Proposed Designs for Snow3G_DES with other Algorithms

3.4. NIST SP 800-22 statistical tests

We performed NIST SP 800-22 statistical tests to test the randomness of 1,000,000 bits of stream keys generated by the two proposed designs and also on 1,000,000 bits of stream keys generated by the ZUC and snow3G algorithms. The results were as shown in Table 3.

Table 3. NIST SP 800-22 Statistical Tests

STATISTICAL TEST	ZUC KEYSTREAM P-VALUE	Snow3G KEYSTREAM P-VALUE	ZUC_AES KEYSTREAM P-VALUE	Snow3G_-DES KEYSTREAM P-VALUE
Frequency		0.739918	0.066882	0.991468
Block Frequency	0.122325	0.534146	0.350485	0.739918
Cumulative Sums		0.911413	0.739918	0.739918
Runs		0.534146	0.534146	0.911413
Longest Run		0.739918	0.350485	0.739918
Rank	0.350485	0.911413	0.534146	0.122325

FFT		0.534146	0.534146	0.350485
NonOverlapping Template	0.350485	0.739918	0.534146	0.534146
Overlapping Template	0.008879	0.991468	0.350485	0.991468
Approximate Entropy	0.017912	0.213309	0.350485	0.122325
Serial	0.534146	0.213309	0.534146	0.534146
Linear Complexity	0.534146	0.350485	0.739918	0.739918
Average	0.2883	0.6178	0.4323	0.6265

Test results for all four algorithms (Table 3) show a significant p-value increase in hybrid models the p-value values in the proposed hybrid designs compared to what they were when ZUC and snow3G were alone. For example, the average p-value for the design ZUC_AES for 12 tests became 0.4323, while it was 0.2883 when it was in the case of ZUC alone, it also almost doubled. We also notice that the second proposed design, snow3G_des, outperforms all the algorithms we tested in the p-value value and in the source [15], meaning that it is the most random among the four tested algorithms. It is worth noting that the test is considered successful if the p-value exceeds 0.01, meaning that all the tests we did were successful.

3.5. Statistics related to the research content

Table 4 shows some of the statistics included in the research

Table 4. Time Research Statistics

Analysis Type	Statistical Indicator	Location in Paper	Value / Result	Source / Reference	Technical Significance
Encryption Time	Average Time	Table 1	ZUC: 0.85s To 1Mbit	Simulation Results	ZUC is significantly faster than all
	Hybrid Model Speed	Table 2	ZUC-AES Model 3: 2.10s	Performance Evaluation	Faster than AES-only; balanced hybrid
	Speed Ratio	Derived from Table 1	ZUC is ~3.15x faster than AES	Own calculation	Indicates performance benefit of stream cipher
Randomness Quality	p-value	Table 3	SNOW3G-DES: p = 0.6265	NIST Tests	Passed randomness evaluation threshold (0.01)
	Average p-value	Table 3	ZUC-AES: p = 0.4323	NIST SP800-22	Acceptable randomness improvement from base ZUC
	Number of NIST Tests Passed	Table 3	All models passed 15/15 tests	SP800-22 Tests	Confirms cryptographic suitability
Security Strength	Key Length		ZUC-AES = 2^{384} Snow3G-DES=2^{320}	Own Equation	Stronger than AES (2^{128}) and SNOW3G (2^{128}) and ZUC

4. Conclusion

By examining our results, we conclude that, In terms of speed. ZUC had the shortest encryption and decryption time among them, followed by Snow3G, followed by block algorithms, led by AES and then DES. For the hybrid cipher we proposed, ZUC-AES, the encryption and decryption time was longer than AES. This is because the number of AES calls in the hybrid design is larger than it would be alone for the same amount of data. The same applies to Snow3G-DES, where the encryption of the proposed design was larger than Des and Snow3G combined for the same reasons. To make it less, we changed the design to make AES or DES calls every certain number of cycles, rather than every cycle, such as 1/4 of the stream key production cycles, or once every two cycles of production, limiting it to just one round. However, this reduces the security of the proposed design because it makes the production of some stream keys dependent only on ZUC or Snow3G. From the results of the NIST SP 800-22 tests, we find that the p-values of the proposed designs (ZUC_AES and snow3G_des) are greater than the single (ZUC and snow3G). If a high level of security is required, then the two basic proposed designs are the best. However, if speed in encryption and decryption is required, then the third proposed design of the two proposed designs, is the best. Security strength increased from 2^{256} (snow3G) to 2^{320} (snow3G-DES) and from 2^{256} in ZUC to 2^{384} in ZUC_AES. The future research should be introducing the RSA algorithm into the two proposed designs, and using it to encrypt AES and Des keys, to significantly increase the security level and ensure secure key exchange between the sender and receiver. And introducing other encryption algorithms into the two proposed designs, such as introducing AES into the snow3G-des design and Des into the ZUC_AES design in the F layer between R_1 and R_2 and between mod and XOR, for example.
Limitations research are: Hybrid models are not suitable for resource-constrained devices (IoT with <512KB of memory) due to the high encryption time. And the encryption time is significantly higher for the two hybrid designs compared to ZUC/SNOW3G alone. To achieve low encryption time for the hybrid designs, the AES and DES cycles in the proposed designs should be reduced, as well as limited to cycles for generating a specific stream key. This impacts security.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding Information

No funding was received from any financial organization to conduct this research

Author Contributions

Ali Ahmed Mustafa contributed to the conceptual design of the study, the design of two algorithms, the development of simulations, and the preparation of the original manuscript draft. Lujain Sabah Abdullah provided supervision and guidance on the methodological structure, critical reviews of the manuscript, and contributed to the editing and final review process. Both authors read and approved the final manuscript.

Acknowledgments

The authors are grateful to the University of Wasit/College of Engineering/Department of Electrical Engineering in Kut, Wasit, Iraq, for their support of this study. They also extend their sincere thanks to the University of Tikrit, College of Engineering, Department of Electrical Engineering.

References

[1] S. K. Patel, S. B. Verma, B. K. Gupta, S. Singh, E. Naresh, and P. K. Pareek, "Advances in authentication and security protocols for 5G networks: a comprehensive survey," *Discover Applied Sciences*, vol. 7, no. 7, 2025.

[2] I. Sahni and A. Kaur, "A systematic literature review on 5G security," *arXiv preprint* arXiv:2212.03299, 2022.

[3] A. Pradhan, S. Das, M. J. Piran, and Z. Han, "A survey on physical layer security of ultra/hyper reliable low latency communication in 5G and 6G networks: recent advancements, challenges, and future directions," *IEEE Access*, vol. 12, pp. 112320–112353, 2024.

[4] E. A. Al-Maqtari and E. A. Al-Maqtari, "Performance evaluation for AES, Blowfish, DES, and 3DES cryptography algorithms," *Partners Universal Innovative Research Publication*, vol. 2, no. 5, pp. 86–95, 2024.

[5] M. Qasaimeh, R. S. Al-Qassas, and M. Ababneh, "Software design and experimental evaluation of a reduced AES for IoT applications," *Future Internet*, vol. 13, no. 11, pp. 1–21, 2021.

[6] A. Sidhu, "Analyzing modern cryptography techniques and reviewing their timeline," *ResearchGate*, pp. 1–10, 2023. [Online]. Available: https://www.researchgate.net/publication/369013746_Analyzing

[7] S. N. Tambe-Jagtap, "A survey of cryptographic algorithms in cybersecurity: From classical methods to quantum-resistant solutions," *Shifra*, vol. 2023, pp. 43–52, 2023.

[8] Z. D. Team, "On the linear distinguishing attack against ZUC-256 stream cipher," *Cryptology ePrint Archive*, Report 2020/1046, pp. 1–8, 2020.

[9] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, and A. Fúster-Sabater, "Analysis and implementation of the SNOW 3G generator used in 4G/LTE systems," *Advances in Intelligent Systems and Computing*, vol. 239, pp. 499–508, 2014.

[10] P. Ekdahl, T. Johansson, A. Maximov, and J. Yang, "A new SNOW stream cipher called SNOW-V," *IACR Transactions on Symmetric Cryptology*, vol. 2019, no. 3, pp. 1–42, 2019.

[11] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2020.

[12] A. H. Awlla and S. M. A. Aziz, "Secure device to device communication for 5G network based on improved AES," *Scientific Journal of Cihan University*, vol. 5, no. 1, pp. 57–67, 2021.

[13] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publication 800-38D, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2007.

[14] B. Sarkar, A. Saha, D. Dutta, G. De Sarkar, and K. Karmakar, "A survey on the advanced encryption standard (AES): A pillar of modern cryptography," *International Journal of Computer Science and Mobile Computing*, vol. 13, no. 4, pp. 68–87, 2024.

[15] R. Kumari, J. G. Pandey, and A. Karmakar, "An RTL implementation of the data encryption standard (DES)," *arXiv preprint* arXiv:2301.05530, pp. 1–10, 2023.

[16] D. A. Silva, D. L. Oliveira, and G. C. Batista, "Design of DES encryption algorithm as bundled-data asynchronous pipeline using FPGA," *Journal of Integrated Circuits and Systems*, vol. 39, pp. 260–270, 2023.

[17] *Data Encryption Standard (DES)*, FIPS PUB 46, Federal Information Processing Standards Publication, National Institute of Standards and Technology, Gaithersburg, MD, USA, 1993.

[18] Z. H. Abdelwahab, T. A. Elgarf, and A. Zekry, "Approved algorithmic security enhancement of stream cipher for advanced mobile communications," *Information Security Journal*, vol. 29, no. 6, pp. 341–365, 2020.

[19] S. Nandi, S. Krishnaswamy, and P. Mitra, "Recent results on some word oriented stream ciphers: SNOW 1.0, SNOW 2.0 and SNOW 3G," in *Information Security and Privacy in the Digital World—Some Selected Topics*, IntechOpen, London, UK, 2022.

[20] R. M. Zaki and H. B. A. Wahab, "A novel SNOW3G-M algorithm and medical," *International Journal of Online and Biomedical Engineering*, vol. 18, no. 3, pp. 134–150, 2022.