JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Standard for Low-Density Parity-Check Codes and Advanced Cryptographic Authentication in 5G systems

[1] **Rahul Sharma,** [2] **Aakanksha S. Choubey,**
[1] M.Tech Research Scholar, [2] Professor
[1,2]Computer Science Engineering
[1, 2] Shri Shankarcharya Technical Campus Bhilai, India

Abstract: Digital communications have always faced significant dependability and security challenges. Latency reduction is one of the most crucial new needs brought forth by the developing 5G technologies. A combined security and reliability solution is put forth in this research to meet both objectives in a single step with lessened computing complexity, which lowers latency for 5G communication systems. The suggested technique simultaneously performs encryption and encoding using the Low-Density Parity Check (LDPC) codes and the Advanced Encryption Standard (AES) cryptosystem. The suggested technique outperforms earlier traditional methods that execute encoding after encryption, according to MATLAB simulation findings. From 2G to 4G, digital communications have seen many developments in recent years. The rapidly expanding needs for communication technology created new expectations that the next 5G network must meet. Reduced latency is one of several modifications that these new applications, however, need. The propagation and switching delays, as well as the computational complexity of the systems, are what generate latency.

IndexTerms **- DES, low-density parity-check (LDPC) code, AES-LDPC Encryption/Encoding, Key Generation and Expansion.**

1. INTRODUCTION

Since the introduction of 2G and 4G digital communications, several developments have taken place. New needs were brought about by the rapidly expanding demands for communication technologies [1]. Three scenarios—ultra-reliable and low-latency communication (URLLC) enhanced mobile broadband (eMBB) and massive machine-type communication (mMTC)—are at the heart of 5G. These three scenarios enable a wide range of applications, including the internet of things (IoT), traffic control and protection, critical industrial and infrastructure applications, and very high-speed data delivery up to 1Gbit/s. The science of encrypting and decrypting data using mathematics is known as cryptography. With the help of cryptography, you may store and send private data in a way that only the intended receiver can read it over public networks like the Internet. to safeguard the data's security.

2. LITERATURE REVIEW

Encryption is used to protect data from being accessed by unauthorized parties. The Data Encryption Standard (DES), a block cypher encryption method, was intended to be replaced with AES in the early 2000s for improved data security. Its "plaintext" input data, which is a fixed length of 16 bytes (128 bits), accepts various encryption keys with lengths of 128 bits, 192 bits, and 256 bits. A key and a non-linear mapping technique known as the S-Box are used to transform the plaintext into cipher text. the plaintext once the key has been added, In each Nr-1 round, the encryption process performs four transformations, including Sub Byte, Shift Row, Mix Column, and Add Round Key, and cancels the Mix Column block in the last round Nr. The initial encryption key is split into sub keys corresponding to each round for AES-128 bits, where Nr = 10 (12 for AES-192 and 14 for AES-256) and Nr = 10 (10 for AES-192 and 14 for AES-256). In the algorithm, sub Bytes handle substitution, while Shift Rows and Mix Columns handle permutation. This action carries out the replacement. Each byte is replaced with another byte in this phase. Sub Bytes is a byte substitution procedure that generates new values by first passing each byte through a separate function. The byte is then transformed into another value by utilizing its hexadecimal code using a table known as the S-box. The nonlinear byte substitution table (S-box) used by the Transformation in the Cypher to process the State acts on each of the State bytes separately. A linear error correcting code, or low-density parity-check (LDPC) code, is a way to send a message across a noisy transmission channel. A sparse Tanner graph (subclass of the bipartite graph) is used to build an LDPC. Applications seeking highly reliable information transfer across bandwidth- or return-channel-constrained networks while contending with erroneous noise are increasingly turning to LDPC codes. LDPC code implementation has lagged behind those of other codes, particularly turbo codes. Gallager made the original discovery of LDPC codes in the early 1960s. In contrast to its earlier codes, such as Turbo codes, LDPC codes later had a significant rediscovery in the late 1990s, during which a significant quantity of literature is being presented to attain greater performance near to the Shannon limit. The National Institute of Standards and Technology (NIST) has released the Data Encryption Standard (DES), a symmetric-key block cypher. A Feistel Cypher is implemented in DES. 16 round Feistel structures are used. Blocks are 64 bits in size. Even though the key length is 64 bits, DES actually uses 56 bits as its actual key length since 8 of the key's 64 bits are only used as check bits. The following graphic shows the general structure of DES. Since DES is based on the Feistel Cypher all that is needed to define it is the round function, key schedule, and initial and final permutation.

3. METHODOLOGY

The joint encryption and error correction techniques outlined in the literature mentioned above struggle to reduce the computing complexity of the system while retaining excellent security and error correction performance. However, the suggested approach may provide high security and strong error-correcting capabilities with minimal complexity, making it appropriate for future communication systems like 5G that demand low latency.

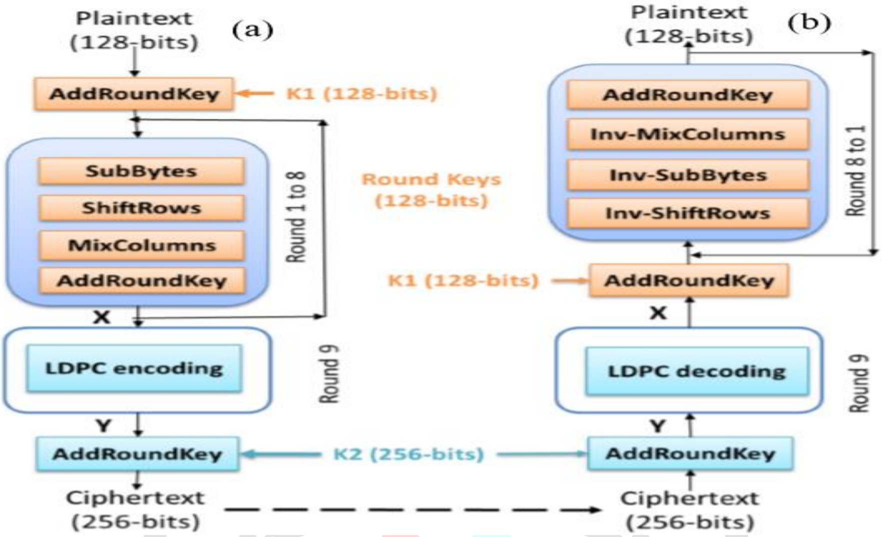


Figure 3.1: The proposed AES-LDPC system' structure

i. AES-LDPC Encryption/Encoding

The goal of this approach is to combine LDPC encoding with AES encryption in order to simplify computations and enhance system performance. The suggested combined AES-LDPC in this article combines AES operations with the LDPC code to encrypt the plaintext as shown in Figure, achieving all the aforementioned properties.

ii. AES-LDPC Decoding/Decryption

The decryption procedure, which may be seen in Fig., is stated as follows and is the reverse of the encryption procedure:

- Ciphertext=Data (256 bits)
- K2⊕Y=Ciphertext
- X=LDPC decoding(Y)
- X K1– 8 1=Plaintext AES (,)

iii. Key Control

Two encryption keys, such as K1 with a length of 128 bits and K2 with a length of 256 bits, are employed in the suggested joint system. Following the LDPC encoding, K2 is used to expand K1 using the usual AES key, w35) as expansion technique for 8 rounds (w0, w1, illustrated in Fig. 8). Since the LDPC codes and the AES algorithm are both openly available, the proposed method's security also relies on the confidentiality of the encryption keys. As a result, K1 and K2 are crucial to the system's security and must be carefully managed by authorized users. In contrast to the conventional AES technique, the suggested solution boosts security by using two encryption keys. Even if an attacker obtains just one encryption key, the decryption procedure, which may be seen in Fig., is stated as follows and is the reverse of the encryption procedure:

Because just one encryption key is utilized in the typical AES technique, the suggested approach has a greater security level.

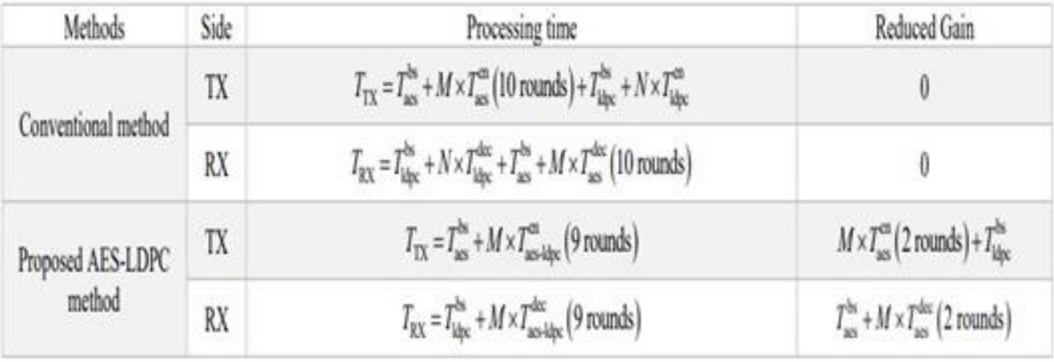
iv. Performance Analysis

The performance of security against cryptanalysis attacks, the efficiency of error correction, and the processing time that define computational complexity are used to demonstrate the gain of the proposed combined AESLDPC technique. The links between encryption and encoding in the overall performances of the system are also demonstrated using numerical figures.

v. Processing Time

Table I, where TX stands for the Transmitter side and RX for the Receiver side, compares the processing times of the proposed technique to the standard method. The AES algorithm runs 8 rounds in the joint method rather than 10 rounds, which provides a reduction in complexity of 2 rounds on both sides. The encoder without code block size calculation (bs Tldpc) receives the encrypted data as input in the ninth round, which uses the LDPC code. In the decoding decryption (bs Taes) portion, the opposite procedure is carried out. When large amounts of data are transferred, more plaintext M are generated and more processing time is gained. Only the tenth round of AES is replaced by the LDPC code in the literature [24], resulting in a twofold slower processing time gain than our approach.

Table 3.1: Processing Time And Reduction Gain



Methods	Side	Processing time	Reduced Gain
Conventional method	TX	$T_{TX} = T_{aes}^{ks} + M \times T_{aes}^{en}(10\,\text{rounds}) + T_{ldpc}^{ks} + N \times T_{ldpc}^{en}$	0
	RX	$T_{RX} = T_{ldpc}^{ks} + N \times T_{ldpc}^{dec} + T_{aes}^{ks} + M \times T_{aes}^{dec}(10\,\text{rounds})$	0
Proposed AES-LDPC method	TX	$T_{TX} = T_{aes}^{ks} + M \times T_{aes\text{-}ldpc}^{en}(9\,\text{rounds})$	$M \times T_{aes}^{en}(2\,\text{rounds}) + T_{ldpc}^{ks}$
	RX	$T_{RX} = T_{ldpc}^{ks} + M \times T_{aes\text{-}ldpc}^{dec}(9\,\text{rounds})$	$T_{aes}^{ks} + M \times T_{aes}^{dec}(2\,\text{rounds})$

vi. Security Performance

The suggested AES-LDPC method's level of security is evaluated by looking at how well-known cryptanalysis methods may break it.

a) Resistance to differential attack

The attempt to determine the encryption key by following the rounds and examining the difference propagation property of cypher pairs is also known as a chosen-plaintext attack [33]. The threshold probability to achieve differential uniformity on a non-linear function of n bits is (1) 2 n Pd , which has the same cryptanalysis difficulty as an O(2n) brute force assault. Two distinct keys, each with a size of n=128 and n=256 bits, are utilized to secure the system in the proposed AES-LDPC system. 1038 and To obtain both K1 and K2, an attacker would need to generate 2128=3.4 2 1077 possibilities, which is still not feasible given today's processing technology (256=1.1, as previously demonstrated).

b) Resistance to saturation or square attack

This uses balance changes rather than an exhaustive search in each round of the AES algorithm to try to estimate the key bytes in the fourth round [32]. Previous research, as mentioned in [24], has demonstrated that higher rounds of AES are safe from this attack on 6 1018 and rounds with a complexity of 263=9.22. Because the proposed system employs 8 rounds of AES encryption, it is protected against saturation attacks. Furthermore, the AES method does not utilize the second key, K2, making it safe against saturation attacks. Additionally, the natural diffusion of the LDPC code dmin with one of the AES 8 rounds is added by the suggested AESLDPC's diffusion complexity. The 256-bit LDPC code's minimum distance is dmin=16 = 42. The diffusion complexity is 49 for the conventional AES-128 (10 rounds) and drops to 47 when 2 rounds are skipped. Therefore, our method's total diffusion is 47 4 2 = 49, maintaining the normal AES128's diffusion complexity. As a result, this technique may withstand a saturation assault in a manner similar to AES with 10 rounds.

vii. Resistance to power analysis attack

In this assault, the perpetrator uses the necessary tools to measure the amount of information that has been compromised throughout the encryption process. Following a statistical examination of power curves, he conjectures the key to unlock the data. The output phase of an AES encryption cycle is where power analysis attacks frequently decide to execute. Based on this, any AES technique utilising a single key, such as in and, may be weak as it may be attacked at the earliest stage, making it simpler to obtain the key. The suggested method is still secure even if an attacker manages to obtain the first AES round key using this way since a brute force assault is required to obtain the second key. A set of tests are run to numerically analyses the security robustness of the method using the following parameters in addition to the security performance analysis in terms of known attacks. Entropy is a measurement of disorder, disarray, and confusion in a picture. Entropy is used to gauge the degree of confusion in the picture data since the primary goal of encryption is to conceal the substance of data. A collection of photos from the database, as seen in Fig. 2, is utilized to verify proper encryption conditions. The results of the entropy calculation on the database's original photos and matching encrypted images are displayed in Table II. According to the findings, all entropy levels are near to the maximum confusion value of 8. As a result, the technique can provide strong protection from entropy analysis.

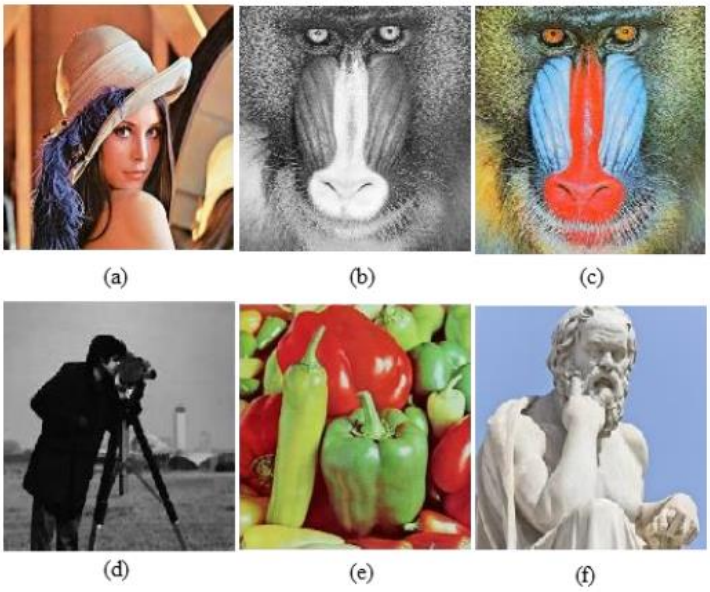
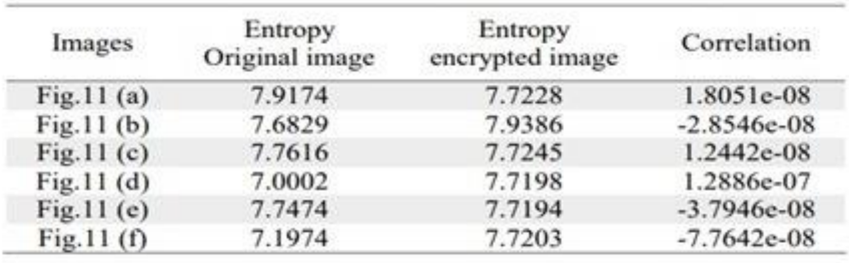


Figure 3.2: Simulation images for entropy and correlation calculations

Table 3.2: Entropy and Correlation Values



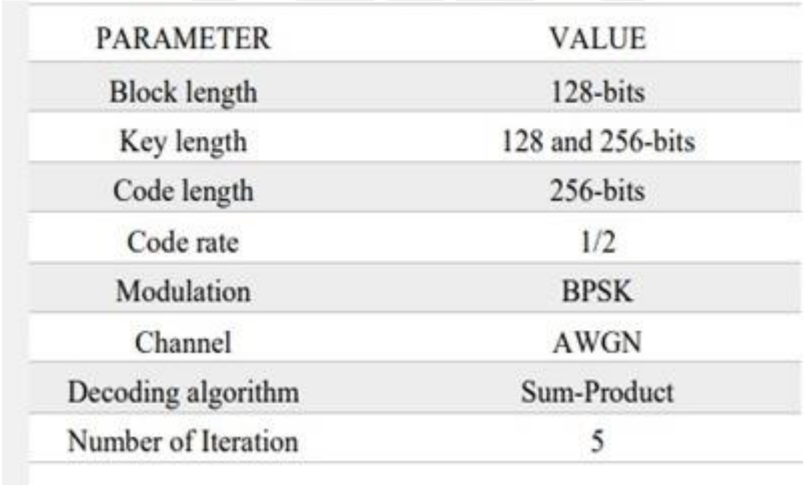
Images	Entropy Original image	Entropy encrypted image	Correlation
Fig.11 (a)	7.9174	7.7228	1.8051e-08
Fig.11 (b)	7.6829	7.9386	-2.8546e-08
Fig.11 (c)	7.7616	7.7245	1.2442e-08
Fig.11 (d)	7.0002	7.7198	1.2886e-07
Fig.11 (e)	7.7474	7.7194	-3.7946e-08
Fig.11 (f)	7.1974	7.7203	-7.7642e-08

An example of a histogram that visualizes the tonal distribution in a digital image is an image histogram. Each tonal value's number of pixels is plotted. A viewer will be able to quickly assess the complete tonal distribution of a picture by taking a quick look at the histogram for that particular image. The distribution of pixels in a picture according to their brightness is shown by the histogram of that image.

4. RESULTS AND DISCUSSION

To test the performances of encryption and error correction, simulations are conducted in MATLAB with the parameters in Table

Table 4.1: Simulation Parameters



PARAMETER	VALUE
Block length	128-bits
Key length	128 and 256-bits
Code length	256-bits
Code rate	1/2
Modulation	BPSK
Channel	AWGN
Decoding algorithm	Sum-Product
Number of Iteration	5

Since the AES-LDPC encoder's coding rate is R=1/2, the input data must be organized in blocks of 128 bits in order to be encrypted with the first key (K1) and produce 256-bit encoded data. Next, the second 256-bit key (K2) is inserted to the 256-bit block (XOR). The data then enters the Additive White Gaussian Noise (AWGN) channel and the Binary Phase Shift Keying (BPSK) modulator, before being sent. After that, the inverse method is used to retrieve the obtained 256-bit data. The following diagram illustrates the simulation outcomes for the combined AES-LDPC approach. The input data for the simulation is the original picture in Fig. 2. As seen in Fig., it is encrypted using the AES industry standard Fig. 3 (a), which was then encrypted and encoded using the suggested AES-LDPC approach. The received picture is recovered and analyzed in accordance with the channel's signal-to-noise ratio (SNR) parameters after transmission across the AWGN channel. When the SNR is 0dB (Fig. 14), the error diffusion effect has completely changed the signal and prevents the image from being recovered; when the SNR is 0.5dB (Fig. 5), the image can hardly be recovered; at 1dB (Fig. 4), users can essentially recognize the image; when the SNR is 1.5dB, the image can be seen by users but still contains some noise; and when the SNR is 2dB (Fig. 5), the image can be seen through which the picture is obtained. The error-correcting performance findings in Fig., however, demonstrate that when SNR is 2.5dB, the image is fully recovered with no mistake. When SNR is equal to or greater than 2dB, the original picture may be seen well with varying SNR values ranging from 0dB to 2.5dB.

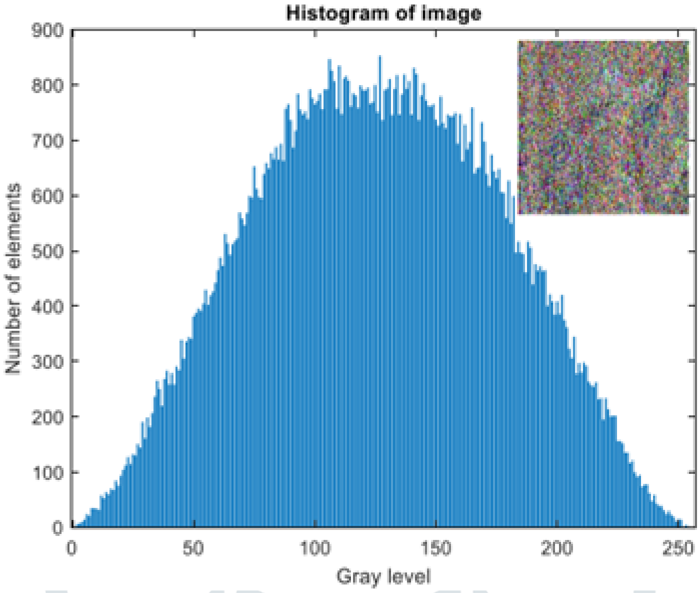


Figure 4.1: Recovered Image at (SNR=0dB)

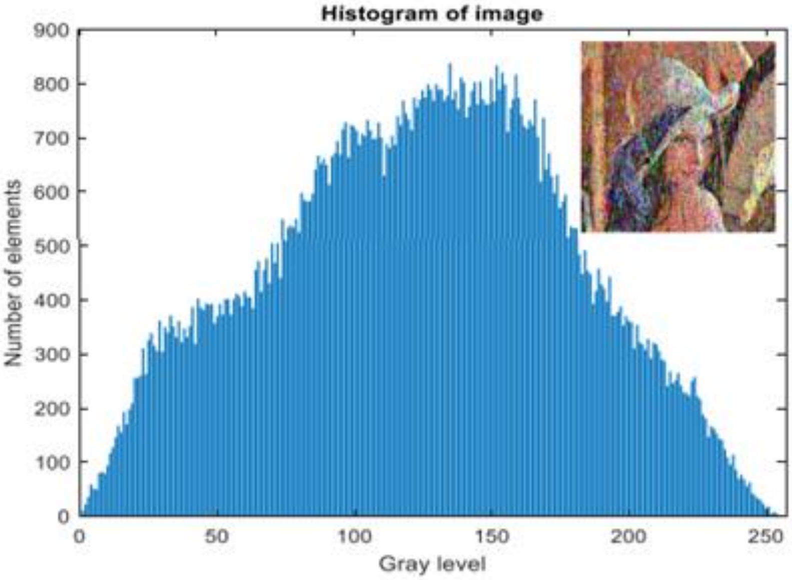


Figure 4.2: Recovered Image at (SNR=0.5dB)

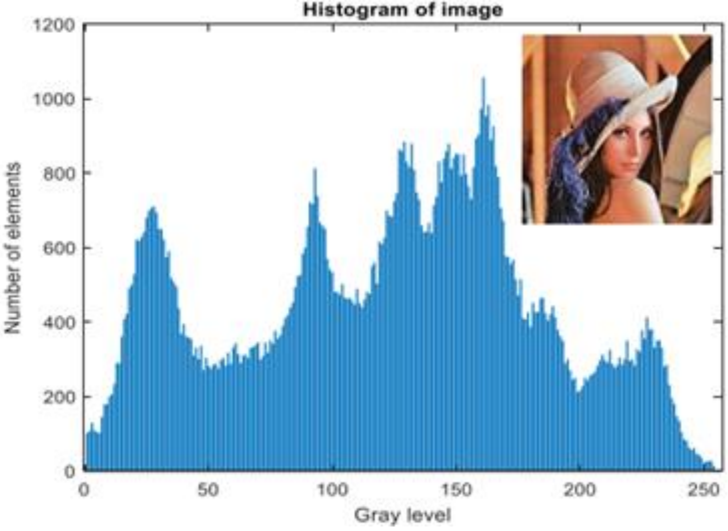


Figure 4.3: Recovered Image at (SNR=2dB)

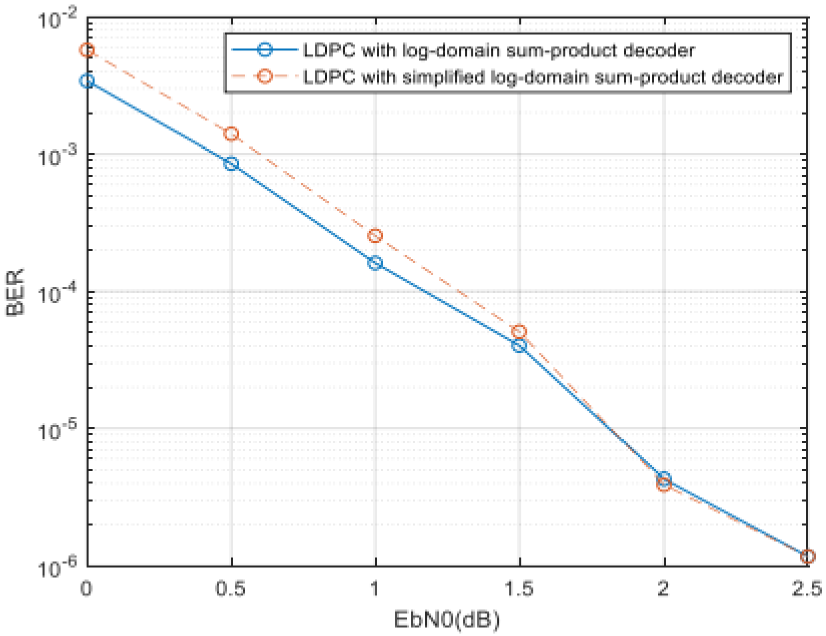
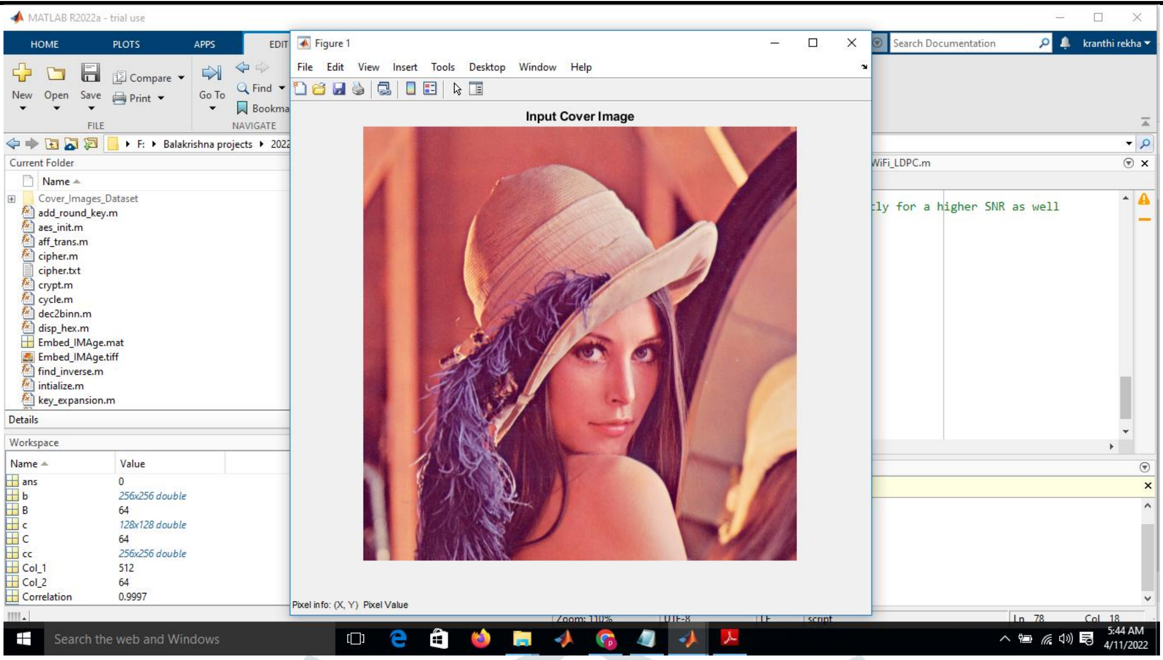


Figure 4.4: The error correcting code



Current Folder

Name
- Cover_Images_Dataset
- add_round_key.m
- aes_init.m
- aff_trans.m
- cipher.m
- cipher.txt
- crypt.m
- cycle.m
- dec2binn.m
- disp_hex.m
- Embed_IMAge.mat
- Embed_IMAge.tiff
- find_inverse.m
- intialize.m
- key_expansion.m

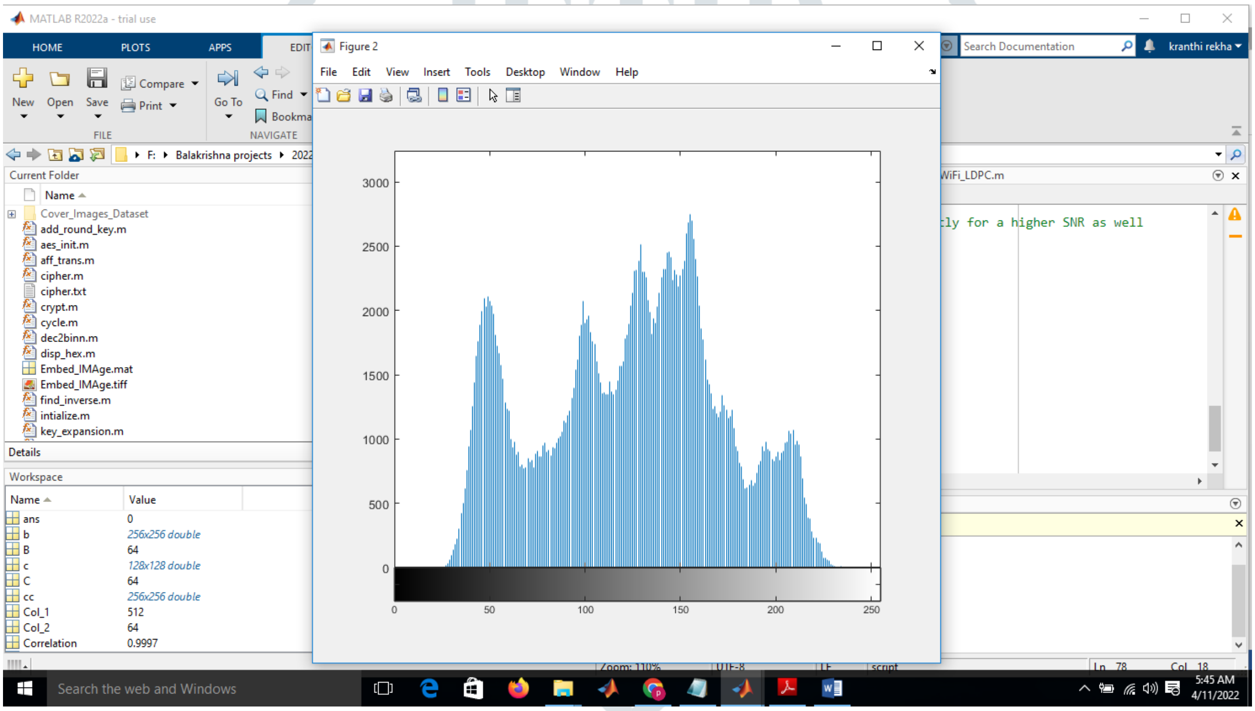
Details

Workspace

Name	Value
ans	0
b	256x256 double
B	64
c	128x128 double
C	64
cc	256x256 double
Col_1	512
Col_2	64
Correlation	0.9997

```
ly for a higher SNR as well
```

Figure 4.5: Input Covered Image



Current Folder

Name
- Cover_Images_Dataset
- add_round_key.m
- aes_init.m
- aff_trans.m
- cipher.m
- cipher.txt
- crypt.m
- cycle.m
- dec2binn.m
- disp_hex.m
- Embed_IMAge.mat
- Embed_IMAge.tiff
- find_inverse.m
- intialize.m
- key_expansion.m

Details

Workspace

Name	Value
ans	0
b	256x256 double
B	64
c	128x128 double
C	64
cc	256x256 double
Col_1	512
Col_2	64
Correlation	0.9997

```
ly for a higher SNR as well
```

Figure 4.6: Histogram View of Input images

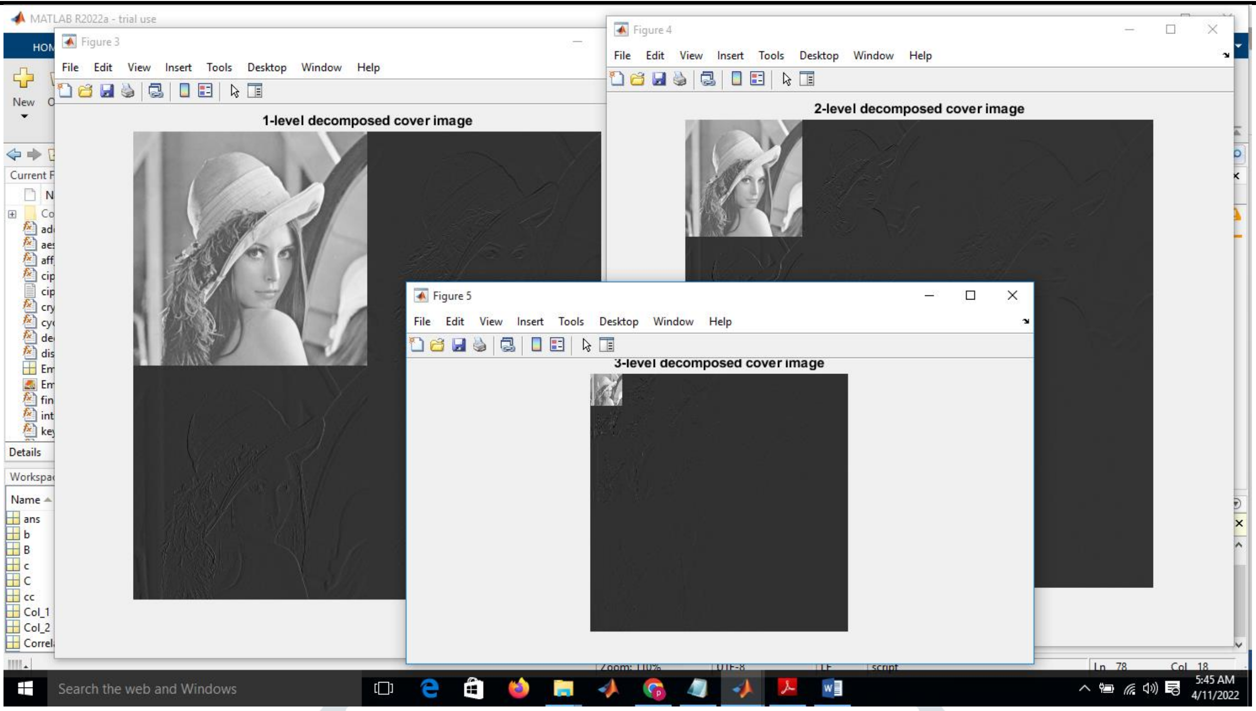
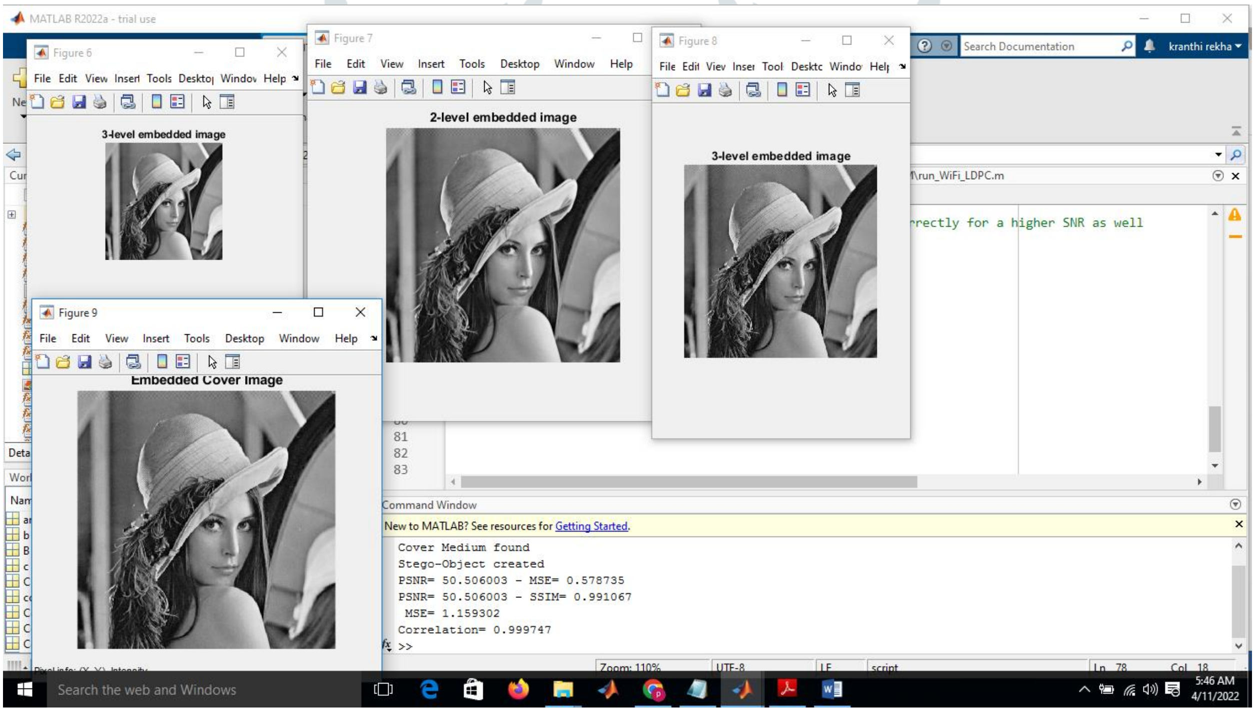
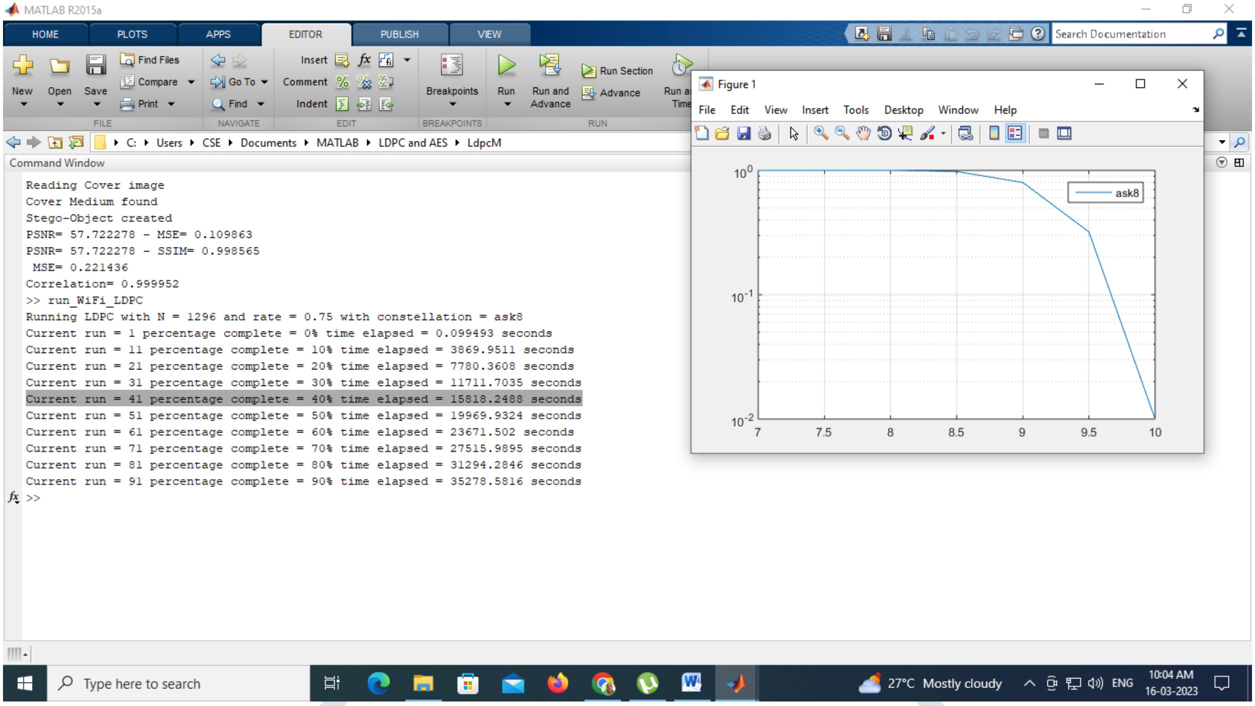

MATLAB R2022a - trial use
Figure 3
1-level decomposed cover image
Figure 4
2-level decomposed cover image
Figure 5
3-level decomposed cover image
File Edit View Insert Tools Desktop Window Help
Search the web and Windows
5:45 AM
4/11/2022

Figure 4.7: 3 Level Decompose Images



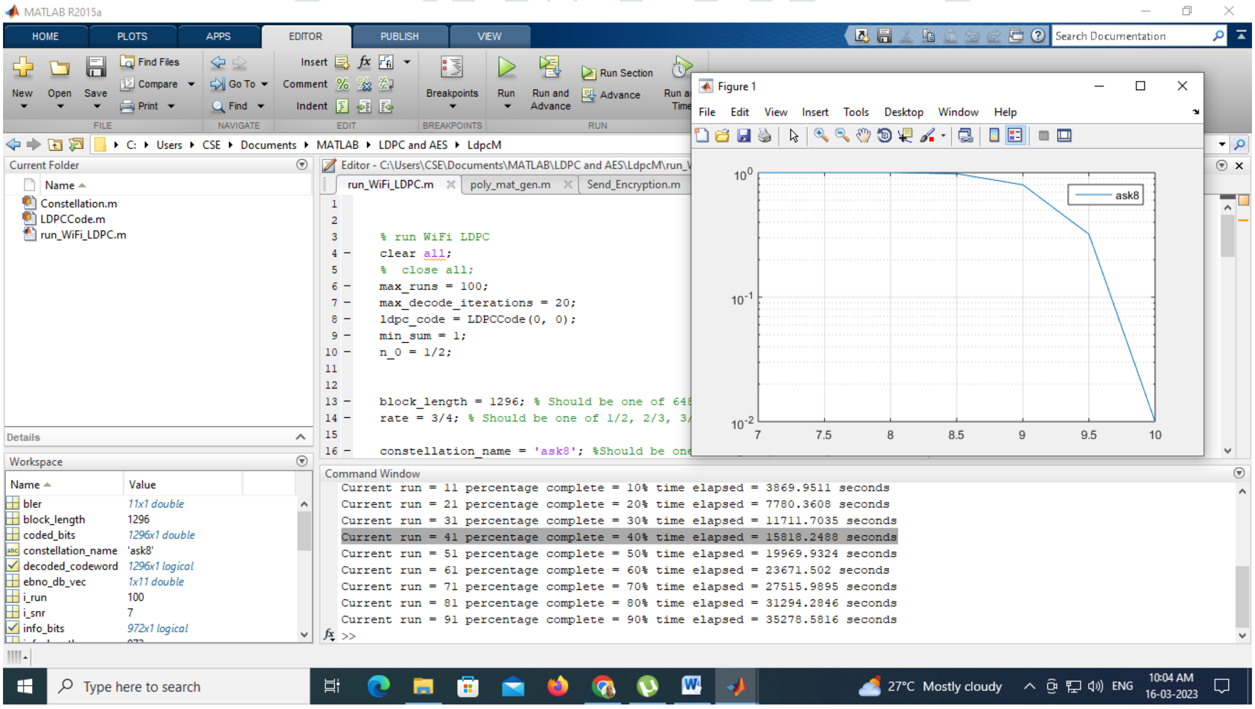
```
Cover Medium found
Stego-Object created
PSNR= 50.506003 - MSE= 0.578735
PSNR= 50.506003 - SSIM= 0.991067
 MSE= 1.159302
Correlation= 0.999747
>>
```

Figure 4.8: 3 level embedded Images



```
Reading Cover image
Cover Medium found
Stego-Object created
PSNR= 57.722278 - MSE= 0.109863
PSNR= 57.722278 - SSIM= 0.998565
 MSE= 0.221436
Correlation= 0.999952
>> run_WiFi_LDPC
Running LDPC with N = 1296 and rate = 0.75 with constellation = ask8
Current run = 1 percentage complete = 0% time elapsed = 0.099493 seconds
Current run = 11 percentage complete = 10% time elapsed = 3869.9511 seconds
Current run = 21 percentage complete = 20% time elapsed = 7780.3608 seconds
Current run = 31 percentage complete = 30% time elapsed = 11711.7035 seconds
Current run = 41 percentage complete = 40% time elapsed = 15818.2488 seconds
Current run = 51 percentage complete = 50% time elapsed = 19969.9324 seconds
Current run = 61 percentage complete = 60% time elapsed = 23671.502 seconds
Current run = 71 percentage complete = 70% time elapsed = 27515.9895 seconds
Current run = 81 percentage complete = 80% time elapsed = 31294.2846 seconds
Current run = 91 percentage complete = 90% time elapsed = 35278.5816 seconds
>>
```

Figure 4.9: MSE output of LDPC



```


% run WiFi LDPC
clear all;
%  close all;
max_runs = 100;
max_decode_iterations = 20;
ldpc_code = LDPCCode(0, 0);
min_sum = 1;
n_0 = 1/2;


block_length = 1296; % Should be one of 648
rate = 3/4; % Should be one of 1/2, 2/3, 3/
 
constellation_name = 'ask8'; %Should be one
```

Workspace

Name	Value
bler	11x1 double
block_length	1296
coded_bits	1296x1 double
constellation_name	'ask8'
decoded_codeword	1296x1 logical
ebno_db_vec	1x11 double
i_run	100
i_snr	7
info_bits	972x1 logical

```
Current run = 11 percentage complete = 10% time elapsed = 3869.9511 seconds
Current run = 21 percentage complete = 20% time elapsed = 7780.3608 seconds
Current run = 31 percentage complete = 30% time elapsed = 11711.7035 seconds
Current run = 41 percentage complete = 40% time elapsed = 15818.2488 seconds
Current run = 51 percentage complete = 50% time elapsed = 19969.9324 seconds
Current run = 61 percentage complete = 60% time elapsed = 23671.502 seconds
Current run = 71 percentage complete = 70% time elapsed = 27515.9895 seconds
Current run = 81 percentage complete = 80% time elapsed = 31294.2846 seconds
Current run = 91 percentage complete = 90% time elapsed = 35278.5816 seconds
>>
```

Figure 4.10: output of AES and LDPC

5. CONCLUSION

To ensure data security and mistake corrections while lowering the computational cost of the system, a combination AES-LDPC technique is suggested. The approach uses two encryption keys to assure security and replaces the last two rounds of the AES cryptosystem with the LDPC code. The suggested approach can withstand all well-known cryptanalysis attacks, according to security analysis. This approach may successfully recover picture data at SNR equal to or greater than 2dB, according to simulation findings across the AWGN channel. The suggested solution maintains the LDPC code's capacity to repair errors while also using its performance to raise the AES algorithm's security level. In comparison to traditional approaches, a large processing time gain is made possible by eliminating the first two rounds of the AES algorithm and cancelling the code block size calculation in the LDPC code. The combined AES and LDPC approach works effectively with less computational complexity without sacrificing each method's performance.

Future work (Scope of work): a joint AES-LDPC method is proposed to achieve both data security and error corrections while reducing the computational complexity of the system. In the method, the LDPC code is embedded in the AES cryptosystem to replace its last two rounds, and two encryption keys are used to ensure security protection. Based on security analysis, the proposed method can resist all the well-known cryptanalysis attacks. Simulation results through the AWGN channel show that this method can accurately recover image data at SNR equal to or greater than 2dB. The error correction capability of the LDPC code is maintained in the proposed method while its performances are used to improve the security level of the AES algorithm. Due to the removed two rounds of the AES algorithm and the code block size determination cancelled in the LDPC code and their parallel activation, a high processing time gain is achieved as compared to the conventional methods. The joint method of AES and LDPC perform well with reduced computational complexity without compromising the performance of each other.

ACKNOWLEDGEMENTS

I am very thankful to my guide and friend who help in research work in this section.

REFERENCES

[1] S. Mattisson, “An overview of 5G requirements and future wireless networks: Accommodating scaling technology,” IEEE Solid-State Circuits Magazine, vol. 10, no. 3, pp. 54–60, 2018.

[2] Q. Xu, D. Gao, T. Li, and H. Zhang, “Low latency security function chain embedding across multiple domains,” IEEE Access, vol. 6, pp. 14474–14484, January 2018.

[3] C. G. Gheorghe, D. A. Stoichescu, and R. Dragomir, “Latency requirement for 5G mobile communications,” in Proc. 10th International Conference on Electronics, Computers and Artificial Intelligence, 2018.

[4] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, “Channel coding for ultra-reliable lowlatency communication in 5G systems,” in Proc. IEEE 84th Vehicular Technology Conference, 2016.

[5] J. H. Bae, A. Abotabl, H. P. Lin, K. B. Song, and J. Lee, “An overview of channel coding for 5G NR cellular communications,” APSIPA Trans. Signal Inf. Process., vol. 8, pp. 1–14, 2019.

[6] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, A Comprehensive Guide to 5G Security, USA: John Wiley & Sons Ltd, 2018.

[7] S. Bhattacharya, “Cryptology and information security-past, present, and future role in society,” Int. J. Cryptogr. Inf. Secur., vol. 9, no. 1, pp. 13–36, 2019.

[8] A. Muhammad Abdullah. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptogr. Netw. Secur. pp. 1–13. [Online]. Available: https://www.researchgate.net/publication/317615794.

[9] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, “Overview of 5G security challenges and solutions,” IEEE Communications Standards Magazine, vol. 2, no. 1, pp. 36– 43, April 2018.

[10] J. Pan, P.-W. Tsai, and J. Watada, Advances in Intelligent Information Hiding and Multimedia Signal Processing, Smart Inno. Springer, 2017.

[11] R. author Ahlswede, A. Ahlswede, I. Althöfer, C. Deppe, and U. Tamm, Hiding Data - Selected Topics : Rudolf Ahlswede’s Lectures on Information Theory 3, Germany: Springer Nature, 2016.

[12] FIPS-197: Specification for the Advanced Encryption Standard (AES), Fed. Inf. Process. Stand. Publ. issued by NIST, 2001.

[13] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard. Berlin: Springer-Verlag Berlin Heidelberg GmbH, 2002.

[14] D. J. C. Mackay and R. M. Neal, “Near shannon limit performance of low density parity check codes-to be published in electronics letters,” Electron. Lett., vol. 33, no. 6, pp. 457–458, 1997.

[15] M. Tomlinson, C. J. Tjhai, M. A. Ambroze, M. Ahmed, and M. Jibril, Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications, Springer Nature.