IJARSE
ISSN 2319 - 8354

A real and accurate Encryption utilizing with compression

Gowtham Mamidisetti^1**, Ramesh Makala**^{2}

^{1}*Associate Professor, Department of Computer Science and Engineering,*

Bhoj Reddy College of Engineering, Hyderabad,

mamidisetti.gowtham@gmail.com

^{2}*Professor, Department of Information Technology,*

RVR&JC College of Engineering, Guntur.

Mail id: mrameshmailbox@gmail.com

Abstract:

The conventional compression mechanisms reducing the size of input information for compression and do not contemplate on additional features. A numerous varieties of compression techniques are in usage and they are working with compression ratios, performance and complexity for the compression process. In this context some algorithms are suitable for high-end applications and others are appropriate for low amount of data. Many encrypted compression algorithms available and having many limitations. Therefore an advanced real time data compression technique is required at the time of encryption with compression. So that in this work present a GoogleNet deep learning mechanism, it is a data encryption and compression technique for real time applications and feasible with above all limitations. Unlike conventional methods of compression, it is only often an encryption technique to secure data, in addition to minimizing data size. The approach GoogleNetis applicable to data of any scale that does not consume any extra memory. This strategy is really simple to encrypt and decrypt and does not require complicated mathematical procedures. Our tests show that it manages fair densityalsoachieves effective encryption under fast time limits. This mechanism is robust technique compared to earlier methods. At last calculating performance measures like accuracy 98.89%, sensitivity 97.86%, F1 score 98.23% and throughput 98.32%, compression ratio 67%.

Keywords: encryption, compression**,** Googlenet and deep learning.

1. Introduction

The data compression platform is offering online applications, social networks, cloud applications, data mining and big data applications. The above all applications and its organizations generate a huge amount of information continuously. This information maintenance and its Encryption is the significant issue to guaranteeing transmission of data through the network. Moreover, Encoding, Decoding problems are currently becoming an important task in digital transactions. Now a day's many users connecting to digital transmission because attracting many entertainment applications, due to this Encryptioncoming into picture. Therefore, an Encoding, Decodingand digital transmission application are required for critical issues.

Data encryption is a technique used to reduce the data size as of size X to size Y, where Y < X is used. There are 2 fundamental kinds of density methods [1]; lossless & lossy. The initial message may be fully preserved from a compressed one in lossless encoding, but certain bits of information are lost in the procedure in lossless compression. We consume introduced the first type of compression in this paper, which consequences in no data loss. Different compression algorithms were developed to minimise data size prior to this work. Through assigning shorter prefix codes to the higher stirring symbol &longer prefix codes to the lowest occurring sign, Huffman Coding[2] reduces scale. Dynamic Huffman Coding[3] tries to solve issues in[2], but dispatcher&recipient essential dynamically create tree in this, which would be a time-consuming operation.

Figure: 1 encryption keys

Run Duration Encoding [4] distinguishes alternating pairs of symbols as well as positions (symbol, N) where N is repeated by the figure of symbols. Through making logical equations, Arithmetic Coding [5] does compression. Efficient Huffman Coding for High Speed Search and Memory [6] increases symbol search speed and mitigates memory size. By defining complement values, Effective Test Pattern density algorithms focused on Complementary Huffman Coding[7] offer improved compression than[2][3].

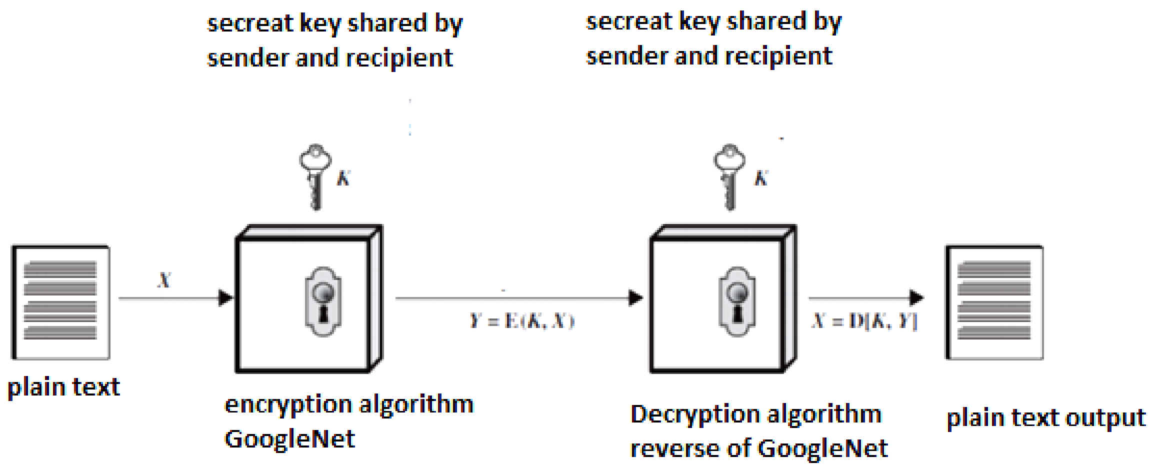
secreat key shared by sender and recipient
secreat key shared by sender and recipient
K
K
X
Y = E(K, X)
X = D[K, Y]
plain text
encryption algorithm GoogleNet
Decryption algorithm reverse of GoogleNet
plain text output

Figure: 2. General encryption method

The initial letter, known as plaintext, is translated here into seemingly random nonsense, known as cipher text. The method of encryption consists of an algorithm and a key. A value independent of the plaintext is the secret. Changing the key alterations is the algorithm's performance[4]. It will be transmitted once the cipher text is created. By utilizing a decryption algorithm and the same key that was used for encryption, the cipher text may be turned back to the original plaintext upon reception. Security is dependent on several variablesByanalysis eight symbols & placing bits of the 8 symbol hooked on the previous 7 bytes, the Authenticated Bit Fluctuating& Stuffing Methodology [8] decreases the data size. But only compression is achieved through these methods and not otherwise. Any of them has drawbacks, functions on such text patterns also is time overriding in execution. However, our suggested solution is ideal for all data sizes, feasible for deployment, and primarily provides protection outside compression.In this way, the remaining organization of the paper follows. Segment 2 gives an impression of the method proposed. A basic definition is seen in Section 3, & Section 4 provides a reverse solution to this strategy. Finally, in section 5, findings are provided.

2. Methodology

The approach suggested recognizes and modifies the compression function in[8] to serve as encryption. It is a symmetrical main encryption process that begins with the sequential formation of blocks of eight bytes in aassumed document. By embedding the eighth byte within the first seven bytes, it attempts to reduce the block size from eight to seven bytes. This is probable since, if we regard ASCII characters, separately uses just 7 bits as an alternative of 8; each character's MSB is then lost. To store a quantifiable terms from other characters, can be use this bit.

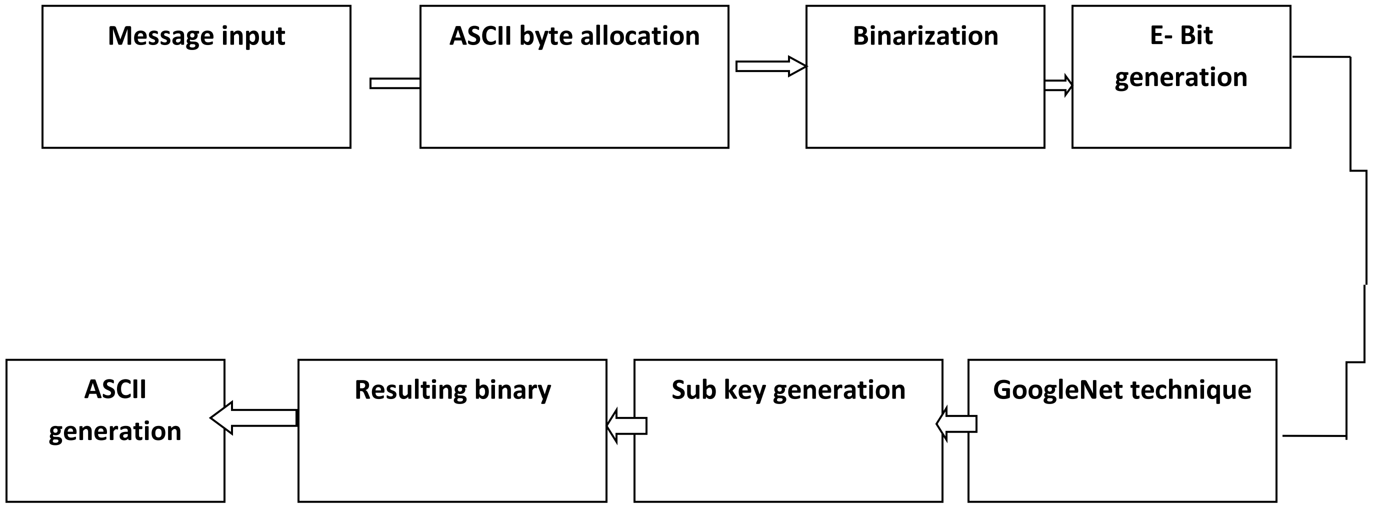

Figure: 3 proposed methodology

Additional characters in [8] are still deposited in MSB bits to have uniformity. But in our system, storage in MSB is not set also we randomly pick positions out of 8. The position chosen randomly has become a subkey but is saved in the key register. It strives to position 0 in locations of 1 and position 1 in locations of 0. to prevent monopoly, Theearlier value of the particular location is then shifted to the consequent character's MSB. For all the resulting blocks, this step is replicated. It significantly improves block values and decreases the size of blocks from 8 to 7 by doing two items at the same time (compression and encryption). Since we store 7 eighth character bits in 8 random positions of the first seven fonts in a block, for each character, we essential a 3 bit key. We then need 21 bits as a sub key for a block and entirely the sub keys for completely the blocks are documented in a key file that is sent to the other side. The suggested algorithm for the scheme is self-explanatory & introduced in Algorithm 1. The output file at the end of the algorithm includes mutually compressed data and encryption keys.

3. Decrypting and decompression

In algorithm 2, the method of having the original message is defined. In order to form a block, it reads seven bytes from the compressed file (also encrypted). It selects a bit from eight possible positions in each block and conducts appropriate interactions and complementation dependent on sub keys in the significant file. Any block generates an eighth byte after this, which is unseen in the first seven bytes. For the remaining blocks, this procedure is replicated and the real message is eventually disclosed.

With such a basic illustration, this segment illustrates the method. Let us understand the "MEDICINE" post. The method of compression & encryption method throughconsequences is seen in table 1. Considering them as 8 bytes fitting to a block, the initial column of the table displays the bytes contained in the text provided. The second & third columns extant ASCII values in addition to the resulting bytes are binary values. According to the method, the eighth byte in each block is embedded in random positions in the first seven bytes of the block. The fourth column then determines which E byte bit is used on behalf of the embedding method. The position where E bit is processed in a byte is randomly chosen and the sub key given in the fifth column has become this random value. When bits are inserted in appropriate bytes, byte values are modified, and are presented in the last two columns of the table with the subsequent binary and ASCII values. E gets inserted in the first seven bytes after the method is done, which decreases the block size between eight to seven. For all contributes are ongoing, such procedure is followed.

Table: 2 experimental results

Byte	ASCII	Binary	E bit	sub key	Resulting Binary	ASCII
E	69	01000101	1	110	11001110	206
N	78	01001110	1	110	11001010	202
G	71	01000111	0	010	10101010	170
I	73	01001001	0	001	01010111	87
N	78	01001110	1	101	11001101	205
E	69	01000101	1	110	00111010	58
E	69	01000101	0	011	10110100	180
R	82	01010010	0	010	01011101	93
I	73	01001001	1	110	01010101	85
N	78	01001110	0	111	10101110	174
G	71	01000111	0	010	00110010	50

Compression earlier encryption, it has extended been recognized that before encrypting, there are benefits of removing uniformities in the plaintext. For the following purposes, compression should be conducted first before encryption,Compressing it last would not decrease the file size much. Effective encryption can render any input data appear random (especially redundant data). But compression works by eliminating redundancy, and on random data, it doesn't work well. Compressing that should minimize any attacks' efficacy. By decreasing the reliability in the files, compression works. Occurrence analysis, which depend on on finding repetitive data, is a common method of cryptanalysis. Compressing it could decrease its potency. Attacks by blunt force can take longer. By attempting subkeysolutions&decrypting the details and verifying whether the output data makes any difference, brute force attacks work. An attacker necessity first decrypt the data by compressing it &then decompress it earlier seeing if the output data kinds any sense. This takes even extended, because if an intruder does not really realise you're compressing the details at all, the code can never be cracked. There is fewer E bit open to adversaries to study. These less details are the rival (intruders) has to examine, the fewer hints they have about your E bit inner representation, and so it's significant.

Performance measures

$$\text{MCC} = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP).(TP+FN).(TN+FP).(TN+FN)}} \quad \text{-------------} \ (2)$$

(MCC: worst value = -1; best value = +1)

$$F_1\ score = \frac{2.TP}{2.TP+FN+FP} \quad \text{-------------------} \ (3)$$

$(F_1\ score : worst\ value = 0; best\ value = 1)$

$$sensitivity = \frac{TP}{TP+FN} \quad \text{--------------------------------} \ (4)$$

$(sensitivity : worst\ value = 0; best\ value = 1)$

$$specificity = \frac{TN}{TN+FP} \quad \text{----------------------------} \ (5)$$

$(specificity : worst\ value = 0; best\ value = 1)$

$$\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}} \ \text{--------} \ (6)$$

$$\text{Space Saving} = 1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}} \ \text{---------} \ (7)$$

$$\text{Data Rate Saving} = 1 - \frac{\text{Compressed Data Rate}}{\text{Uncompressed Data Rate}} \ \text{------} \ (8)$$

In this research work we are analyzing the various methodologies of compression and encryptionmethods,. Various encryption methodologies such as AAS, MD5, RAS, TWOFISH and TripleDES are providing network security and solves the many real time problems, but the dynamic attacks are cannot handle compression, encryption with quick time. Therefore an advanced compression and encryption technique is required for network security, this possible only with deep learning mechanism, so GoogleNet deep learning cryptography algorithm is proposed for dynamic network applications. It is a dynamic steps and variable key size method. This method solves the current network issues and providing mean error 0.19, objective function 0.98, sensitivity 0.97, accuracy 0.98 and F1 score 0.98 have been attained. These experimental results are competing with earlier methods and outperform the methodology.

Next, the encryption algorithm must be efficient enough that decrypting a message on the basis of cipher text alone is impractical. Moreover, confidentiality relies on the key's confidentiality, not the algorithm's secrecy show in fig.5.

Table: 3 performance analysis

	Mean error	Objective function Estimation	Sensitivity	Accuracy	F1-score
AES	0.512	0.8925	0.658	0.489	0.89
MD5	0.428	0.812	0.684	0.387	0.91
RSA	0.321	0.719	0.658	0.289	0.899
Twofish	0.25	0.912	0.76	0.458	0.954
TripleDES	0.21	0.8759	0.787	0.21	0.915
GoogleNet	0.19	0.984	0.9786	0.9889	0.9823

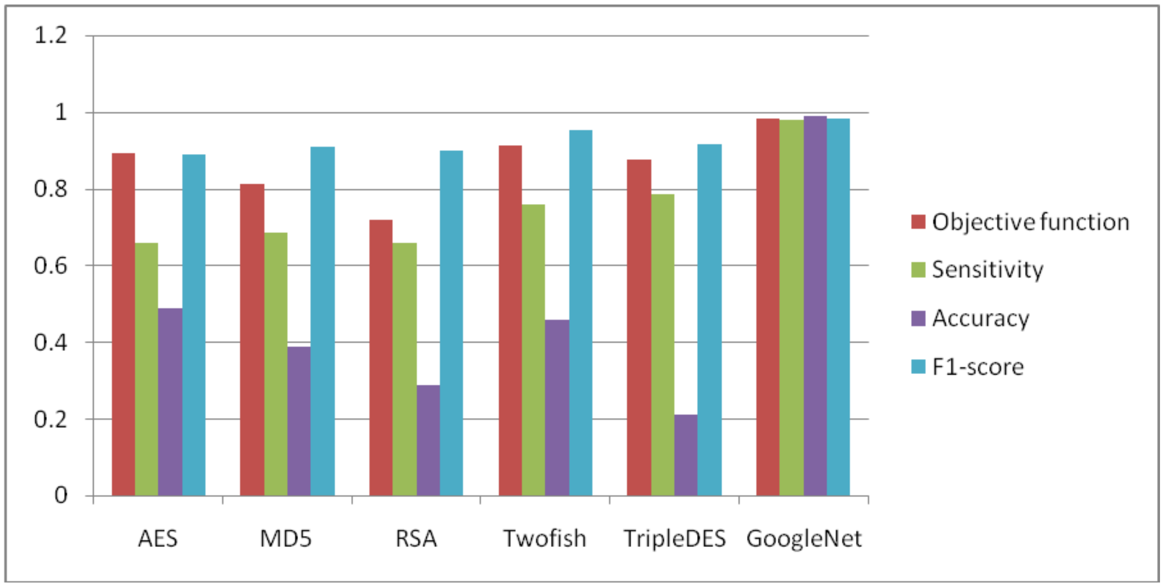

Figure: 6 results analysisigure 6 and table 3 explains about performance analysis of measures, in this mean square error, objective function, sensitivity, accuracy and F1 score is calculated. At all conditions proposed method attains more improvement.

Table: 4 experiment results analysis

Method	**Compression Ratio %**	**Space Saving %**	**Data Rate Saving %**
[10]	28	10	11
[11]	34	14	14
[12]	42	19	17
[13]	51	24	19
Proposed	67	31	22

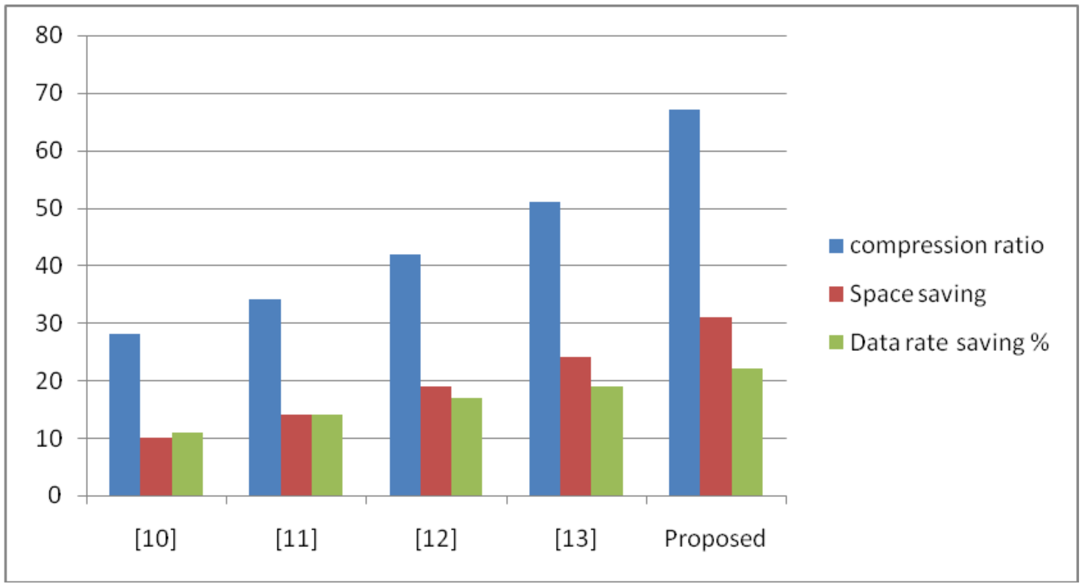

Figure: 6final results

Table: 4 and fig 6 clearly explains about graphical representation of various metrics. In this Compression ratio, space saving and Data rate saving elements are improved by proposed method.

Conclusion

In this research work data compression with encryption model is proposed with Googlenet technique. Earlier methods attains low encryption and compression accuracy, many conventional models are not suitable for current technology. Proposed model is working with deep learning mechanism so compression with encryption has been done with accurate manner. At last calculating performance measures like accuracy 98.89%, sensitivity 97.86%, F1 score 98.23% and throughput 98.32%, compression ratio 67%. The space saving and data rate saving is improved by 31% and 22% respectively. Implemented model is compute with existed methods and outperformance the accuracy.

REFERENCES

[1] "Introduction to Data Compression", Khalid Sayhood, Morgan Kaufmann, 1996.
[2] "A Method for the Construction of Minimum – Redundancy codes", Proceedings of the IRE, sept 1952, 1098 – 1102.
[3] http://en.wikipedia.org/wiki/DynamicHuffmanCoding
[4] http://en.wikipedia.org/wiki/Information_Theory
[5] http://en.wikipedia.org/Arithmetic_Coding

[6] "Memory Efficient and High Speed Search Huffman Coding", Reza Hashemian, IEEE Transactions on Communications, Vol. 43, no 10, Oct 1995, 2576-2581

[7] "Efficient Test Pattern Compression Techniques based on Complementary Huffman Coding", Shyue-Kung Lu et al, IEEE 2009.

[8] "An Authenticated Bit Shifting and Stuffing (BSS) Methodology for Data Security", B. Ravi Kumar et al, Computer Engineering and Intelligent Systems, vol 2, no 3, 94 – 104.

[9] DemijanKlinc_, Carmit Hazayy, Ashish Jagmohan, Hugo Krawczyk and Tal Rabin ,"On Compression of Data Encrypted with Block Ciphers", http://users.ece.gatech.edu/~demi/docs/dcc09pap er.pdf.

[10] W. Mao, Modern Cryptography: Theory and Practice. Prentice Hall, 2003.

[11] ChuanfengLv, QiangfuZhao,"Integration of Data Compression and Cryptography: Another Way to Increase the Information Security". AINA Workshops (2) 2007: 543-547

[12] Sien, O.B, Samsudin, A., Budiarto, R." A new image-database encryption based on a hybrid approach of data-at-rest and data-inmotion encryption protocol",Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference.

[13] M. Johnson, D. Wagner, and K. Ramchandran, \On compressing encrypted data without the encryption key," in Proc. of the Theory of Crypto. Conf., Cambridge, MA, Feb.2004.

[14] William Stallings. Advanced Encryption Standard. Chapter 5 in Cryptography and Network Security: Principles and Practices. International Third Edition. Prentice Hall: United States of America. (2003)

[15] Application Security, Inc. Encryption of Data at Rest - Database Encryption. White Paper. (2002)

[16] Lala Krikor, Sami Baba, ThawarArif, Zyad Shaaban," Image Encryption Using DCT and Stream Cipher", European Journal of Scientific Research, ISSN 1450-216X Vol.32 No.1 (2009), pp.47-57.