ISSN:0975-9646

Security Enhanced Communication Scheme with Error Correction Capability and Efficient Channel Utilization

Poornima.P.V[1], Amrutha.V[2].

[1]*Student,* [2]*Assisant Professor,*
Department of Electronics and Communication Engineering,
Sathyabama University
Jeppiar Nagar, Rajiv Gandhi Salai, Chennai-600119

***Abstract*—Security,channel utilization and error correction are of great importance especially in the case of any bandlimited channels which demands high security.This paper describes a new communication scheme which provide enhanced security without compromising on channel utilization and has capability of correcting bit errors.Traditionaly messages were compressed then subjected to encryption before transmission but here the message is encrypted and then compressed,the encryption scheme being employed is DES algorithm.By compressing the encrypted data using RLE,more data can be send through the channel whereby can achieve efficient channel utilization without compromising on the security of the encryption scheme. For combating bit errors,an EDAC system is developed which make use of Hamming codes which can detect and correct one bit errors effectively.When a corrupted message is received,then it is corrected by the error correction scheme.So, the error correction plays an important role in the overall efficiency of the proposed scheme. The overall system can achieve improved detection accuracy, better channel utilization and provide highly secure and reliable communication.The entire system is functionally simulated and verified using Xilinx 9.1 and Modelsim 6.1.**

***Keywords*— EDAC system,DES algorithm,RLE,Hamming codes**

I. INTRODUCTION

Our modern world teems with communication.In modern commuication world Security,Channel Utilization and Error Correction are of great importance when bandwidth limited channels are considered which demands high security..In traditional schemes,the data is compressed,encrypted and then encoded for transmission;which are prone to bit errors while transmission and also poses a security risk as the compressed and encrypted data are susceptible to statistical analysis based decryption mechanisms.Due to these reasons,the existing schemes are not very reliable and usually encrypted data is sent without any compression but this leads to poor channel utilization

A new communication scheme which can provide enhanced security without compromising on channel utilization and has capability of decting and correcting errors is developed.Here, we use encryption and compression scheme in which we use DES algorithm to achieve enhanced security while making it suitable for compression using Run Length Encoding (RLE). By compressing the encrypted data, we can send more data through the channel whereby we can achieve efficient channel utilization without compromising on the security of the encryption scheme. For combating bit errors, we use an Hamming code which can efficiently detect and correct bit errors. The receiver that receives a corrupted form of transmitted word which is corrected by the error correction scheme and then jointly decompressed and decrypted. So, the error correction plays an important role in the overall efficiency of the proposed scheme. The overall system can achieve improved detection accuracy, better channel utilization and provide highly secure and reliable communication. The proposed scheme can be used in all bandwidth limited channels which demands high security.

The organization of the this paper is explained herefore, Section II defines the problem existing,summarizes existing work on the subject and the solution for thr problem. Section III concentrates on DES algorithm and explain how it works.Section IV shows how compression is enabled using the RLE scheme, while Section V covers the need of security and how EDAC systems helps to maintain security for a transmitted message.Section VI presents some simulation results and discussions and finally, Section VII concludes the paper

II. PROBLEM DEFINITION

The real problem arised when the on compressed encrypted data was ready for transmissions.The number of problems arised which results in errors. Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference.This interference can change the shape of the signal.In a single-bit error, a 0 is changed to a 1 or vice versa. The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or vice versa.The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.So here arises a need for error correction and detection in the system to enhance the efficiency of our communication system. The correction of errors is more difficult than the detection.In error detection,the prime importance to see if any error has occurred. The answer is a simple yes or no.

The interest is not in the number of errors. A single-bit error is the same as a burst error. In error correction, there arises a need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors

In traditional schemes, the data is compressed, encrypted and then encoded for transmission; which are prone to bit errors while transmission and also poses a security risk as the compressed and encrypted data are susceptible to statistical analysis based decryption mechanisms. Due to these reasons, the existing schemes are not very reliable and usually encrypted data is sent without any compression but this leads to poor channel utilization

Both encryption and compression are two important systems that enable to improve the communication system,but which should be preceeding the other was a question of great importance.Traditionally compression was carried out first then encryption.If it was done in the other order,ie: encryption then compression,the data after encryption will be highly random(higher the order of randomness higher the order of security) then while considering compression of the encrypted data,it will become meaning less,as random data are less susceptible to compression

There exist scenarios where there is a need to reverse the order in which data encryption and compression are performed. For example, consider a network of low-cost sensor nodes that transmit sensitive information over the internet to a recipient. The sensor nodes need to encrypt data to hide it from potential eavesdroppers, but they may not be able to perform compression as that would require additional hardware and thus higher implementation cost. On the other hand, the network operator that is responsible for transfer of data to the recipient wants to compress the data to maximize the utilization of its resources. It is important to note that the network operator is not trusted and hence does not have access to the key used for encryption and decryption. If it had the key, it could simply decrypt data, compress and encrypt again.It is theoretically possible to compress the source to the same level as before encryption. However, in practice, encrypted data appears to be random and the conventional compression techniques do not yield desirable results. It was long believed that encrypted data is practically incompressible. A surprising paper breaks that paradigm and shows that the problem of compressing one-time pad encrypted data translates into the problem of compressing correlated sources, which was solved by Slepian and Wolf and for which practical and efficient codes are known. Compression is practically achievable due to a simple symbol-wise correlation between the key (one-time pad) and the encrypted message. However, when the correlation is more complex, as in the case of block ciphers, the approach to Slepian–Wolf coding utilized is not directly applicable.Before moving on to the details of the project let us overview the existing technologies and techniques that can channel our work.

Through this piece of work,a new communication scheme have been proposed which can provide enhanced security without compromising on channel utilization and has capability of correcting bit errors. Here a combined encryption and compression scheme in which DES algorithm.By compressing the encrypted data,more data can be send through the channel whereby can achieve efficient channel utilization without compromising on the security of the encryption scheme. For combating bit errors, Hamming codeis being used which can detect two bit errors and correct one bit erroe effectively The receiver receives a corrupted form of transmitted word which is corrected by the error correction scheme and then decompressed and decrypted. So, the error correction plays an important role in the overall efficiency of the proposed scheme. The overall system can achieve improved detection accuracy, better channel utilization and provide highly secure and reliable communication. The proposed scheme can be used in all bandwidth limited channels which demands high security.

In proposed system we are introducing a error detection and correction into the existing system that it enhances the error correcting capability of the system.As discussed before the error correctivity is as important as data security and channel utilization.We are introducing the error detection and correction mechanism using a Hamming encoder decoder system,both which act as a flow through system.

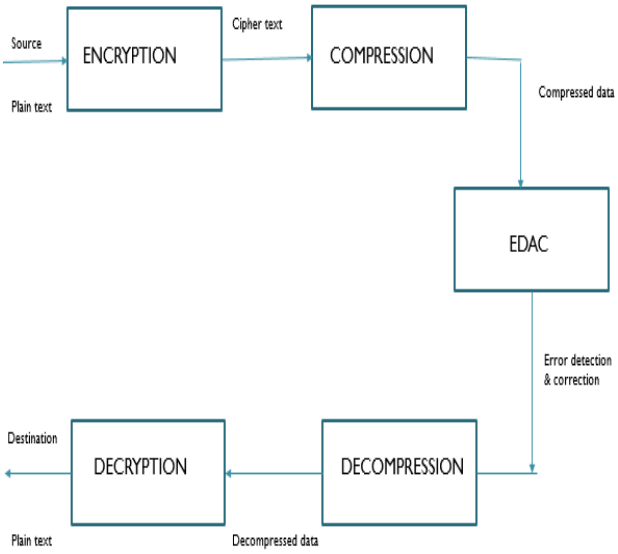

Fig 1:Proposed Syste

III. DES ALGORITHM

The Data Encryption Standard (DES) is a block cipher that uses shared secret encryption. It was selected by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976 and which has subsequently enjoyed widespread use internationally. It is based on a symmetric-key algorithm that uses a 56-bit key. The algorithm was initially controversial with classified design elements, a relatively short key length, and suspicions about a National Security Agency (NSA) backdoor. DES consequently came under intense academic scrutiny which motivated the modern understanding of block ciphers and their cryptanalysis.

Before moving into the details of des algorithm basics of cryptography.It includes two basic components: Encryption algorithm and Keys. If sender and recipient use the same key then it is known as symmetrical or private key cryptography. It is always suitable for long data streams. Such system is difficult to use in practice because the sender and receiver must know the key. It also requires sending the keys over a secure channel from sender to recipient. The question is that if secure channel already exist then transmit the data over the same channel.On the other hand, if different keys are used by sender and recipient then it is known as asymmetrical or public key cryptography. The key used for encryption is called the public key and the key used for decryption is called the private key. Such technique is used for short data streams and also requires more time to encrypt the data . To encrypt a message, a public key can be used by anyone, but the owner having private key can only decrypt it. There is no need for a secure communication channel for the transmission of the encryption key. Asymmetric algorithms are slower than symmetric algorithms and asymmetric algorithms cannot be applied to variable-length streams of data.

There are two techniques used for data encryption and decryption, If sender and recipient use the same key then it is known as symmetrical or private key cryptography. It is always suitable for long data streams. Such system is difficult to use in practice because the sender and receiver must know the key. It also requires sending the keys over a secure channel from sender to recipient.There are two methods that are used in symmetric key cryptography: block and stream.The block method divides a large data set into blocks (based on predefined size or the key size), encrypts each block separately and finally combines blocks to produce encrypted data. The stream method encrypts the data as a stream of bits without separating the data into blocks. The stream of bits from the data is encrypted sequentially using some of the results from the previous bit until all the bits in the data are encrypted as a whole.

If sender and recipient use different keys then it is known as asymmetrical or public key cryptography. The key used for encryption is called the public key and the key used for decryption is called the private key. Such technique is used for short data streams and also requires more time to encrypt the data.Asymmetric encryption techniques are almost 1000 times slower than symmetric techniques, because they require more computational processing power.To get the benefits of both methods, a hybrid technique is usually used. In this technique, asymmetric encryption is used to exchange the secret key; symmetric encryption is then used to transfer data between sender and receiver.

The algorithm's overall structure is shown in Fig.2: there are 16 identical stages of processing, termed rounds. There is also an initial and final permutation, termed IP and,FP, which are inverses (IP "undoes" the action of FP, and vice versa). IP and FP have no cryptographic significance, but were included in order to facilitate loading blocks in and out of mid-1970s 8-bit based hardware.[22]Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme. The Feistel structure ensures that decryption and encryption are very similar processes the only difference is that the subkeys are applied in the reverse order when decrypting. The rest of the algorithm is identical. This greatly simplifies implementation, particularly in hardware, as there is no need for separate encryption and decryption algorithms.The $\oplus$ symbol denotes the exclusive-OR (XOR) operation. The F-function scrambles half a block together with some of the key. The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round. After the final round, the halves are swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.

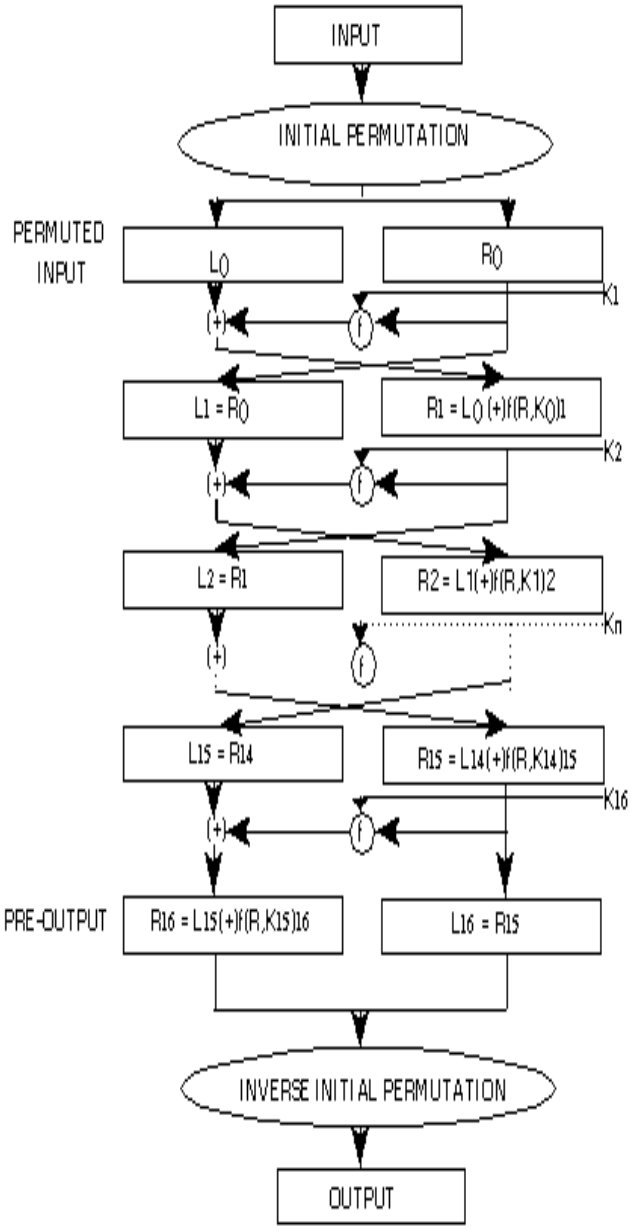

Fig 2: DES Fiestel diagram.

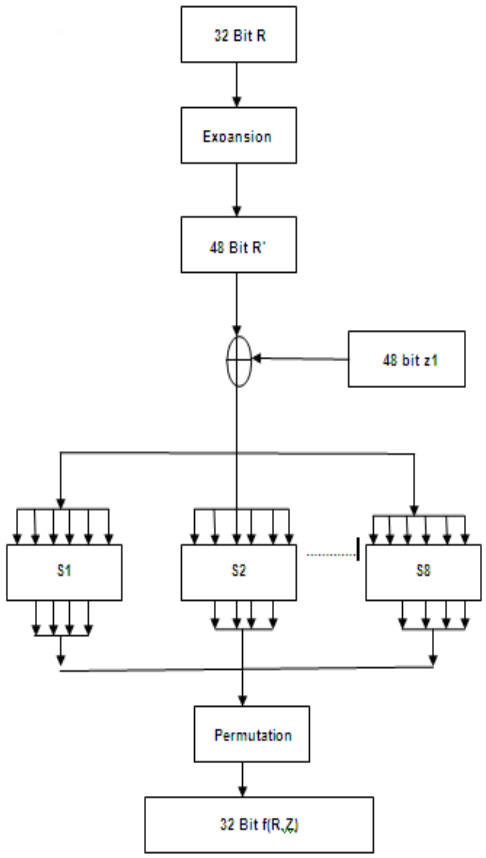

Fig 3:F function

The Feistel function (F-function) of DES. The F-function, depicted in Figure3, operates on half a block (32 bits) at a time and consists of four stages:

Expansion — the 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit (8 * 6 = 48 bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

Key mixing — the result is combined with a subkey using an XOR operation. 16 48-bit subkeys one for each round are derived from the main key using the key schedule (described below).

Substitution — after mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES — without them, the cipher would be linear, and trivially breakable.

Permutation — finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. This is designed so that, after permutation, each S-box's output bits are spread across 4 different S boxes in the next round.

The alternation of substitution from the S-boxes, and permutation of bits from the P-box and E-expansion provides so-called "confusion and diffusion" respectively, a concept identified by Claude Shannon in the 1940s as a necessary condition for a secure yet practical cipher.

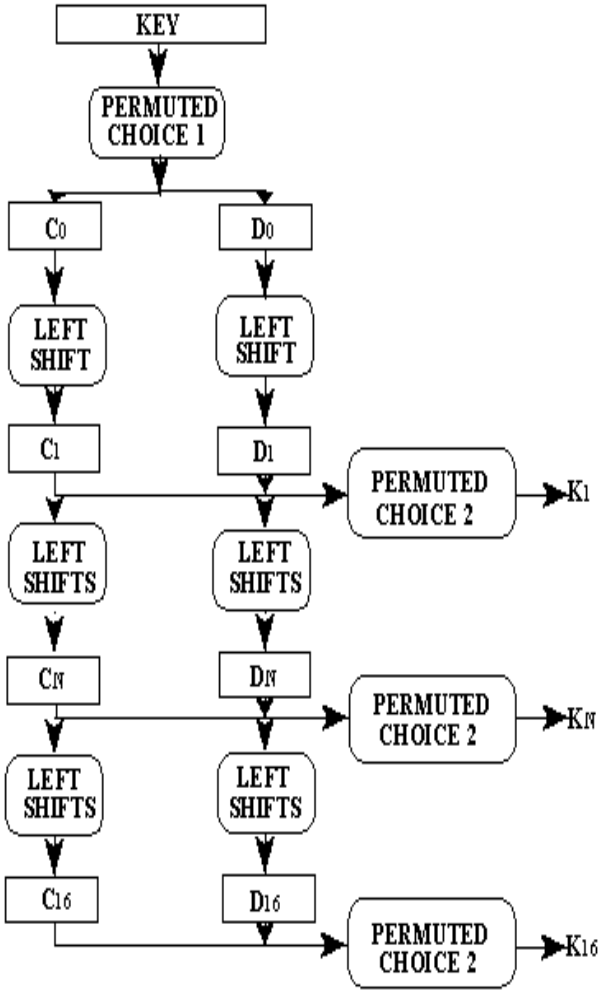

Fig 4:Key schedule calculation

Figure 4 illustrates the key schedule for encryption the algorithm which generates the subkeys. Initially, 56 bits of the key are selected from the initial 64 by Permuted Choice 1 (PC-1) the remaining eight bits are either discarded or used as parity check bits. The 56 bits are then divided into two 28-bit halves; each half is thereafter treated separately. In successive rounds, both halves are rotated left by one or two bits (specified for each round), and then 48 subkey bits are selected by Permuted Choice 2(PC-2) 24 bits from the left half, and 24 from the right. The rotations (denoted by "<<<" in the diagram) mean that a different set of bits is used in each subkey; each bit is used in approximately 14 out of the 16 subkeys.The key schedule for decryption is similar the subkeys are in reverse order compared to encryption. Apart from that change, the process is the same as for encryption. The same 28 bits are passed to all rotation boxes.

IV DATA COMPRESSION

Data compression or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although its formal name is source coding (coding done at the source of the data before it is stored or transmitted).

Compression is useful because it helps reduce resource usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression; this situation is far from being a free lunch. Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (e.g., when using lossy data compression), and the computational resources required to compress and uncompress the data.

There are two types of data compressions**.**Lossless data compression algorithms usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-world data has statistical redundancy. For example, an image may have areas of colour that do not change over several pixels; instead of coding "red pixel, red pixel," the data may be encoded as "279 red pixels". This is a basic example of run-length encoding; there are many schemes to reduce file size by eliminating redundancy.

Lossy data compression is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. Lossy data compression schemes are informed by research on how people perceive the data in question. For example, the human eye is more sensitive to subtle variations inluminance than it is to variations in color. JPEG image compression works in part by rounding off nonessential bits of information. There is a corresponding trade-off between preserving information and reducing size. A number of popular compression formats exploit these perceptual differences, including those used in music files, images, and video.Lossy image compression can be used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the lossy MPEG-2 Video codec for video compression.Inlossy audio compression, methods of psychoacoustics are used to remove non-audible (or less audible) components of the audio signal. Compression of human speech is often performed with even more specialized techniques; speech coding, or voice coding, is sometimes distinguished as a separate discipline from audio compression. Different audio and speech compression standards are listed under audio codecs. Voice compression is used in Internet telephony, for example audio compression is used for CD ripping and is decoded by audio players.

RLE is a very simple form of data compression in which runs of data (that is,sequences in which the same data value occurs in many consecutive data elements) are stored as a single datavalue and count, rather than as the original run. This is most useful on data that contains many such runs: forexample, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don'thave many runs as it could greatly increase the file size.RLE may also be used to refer to an early graphics file format supported by CompuServe for compressingblack and white images, but was widely supplanted by their later Graphics Interchange Format.Typical applications of this encoding are when the source information comprises long substrings of the same character or binary digit.

The actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same.Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space.In our project we are going to make use RLE compression scheme.

In RLE the redundant runs of the data is taken into consideration.The redundant runs are then compressed and send along the transmission line.It can also be done in another way the data bits are covered up by zero bits for transmission.The logic used here is the no voltage is required for the transmission of zeroes.

The algorithm used for both the compressor and decompressor units are discussed here.

Compressor

Step 1:Get four bits

Step 2:Check whether the bits are equal

Step 3:If equal (1111)then send ‘zeroes’ in corresponding positions.

Step 4:If all four bits are zeroes then also send zeroes.

Step 5: Mark the position of all ones and zeroes.

Decompressor

Step1:Again group the incoming data in to group of four bits.

Step 2:Note the positions where it is zeroes

Step 3: Counter check with the positions of all ones and zeroes noted

Step4:In places of all ones again put original values,and send zeroes as such

Step5: The decompressed output will be obtainted

V. EDAC SYSTEM

The error detection and correction (EDAC) are carried out by hamming codes.Here the incoming 64bit data is transmitted by appending 7 parity bits along the 64 bits.The parity bits are generated using the XOR ing of the data bits.The signal transmitted consisted of 64bit data and 7 bit

parity,If the data gets corrupted due to the transmission loss or any electrical discharges with the help of syndrome masking technique the data is corrected and the corrected output is obtained at the output

The correction of errors is more difficult than the detection.In error detection, we are looking only to see if any errorhas occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error isthe same for us as a burst error. In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.To calculate the number of redundancy bits r required to correct a given number of data bits m. we must find a relationship between m and r. With m bits of data and r bits of redundancy added to them, the length of the resulting code is m + r. If the total number of bits in a transmittable unit is m + r. then r must be able to indicate at least m+r+1 different states. Of these, one state means no error,and m + r states indicate the location of an error in each of the m + r positions.So m+r+1 states must be discoverable by r bits: and r bits can indicate 2r different states. Therefore 2r must be equal to or greater than m + r + 1:2 => + + 1For example, if the value of m is 7 (as in a 7-bit ASCIIcode), the smallest r value that can satisfy this equation is4:24 =>7 + 4+l

Hamming provides a practical solution. The Hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits discussedabove. For example, a 7-bit ASCII code requires 4 redundancy bits that can be added 10 the end of the data unit or interspersed with the original data bits. In Fig 1.8 these bits are placed in positions 1, 2 ,4, and 8 (the positions in an 11-bit sequence that are powers of 2). For clarity in the examples below, we refer to these bits as r1, r2,r4, and r8.

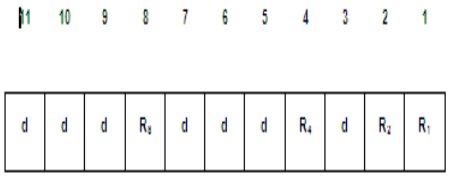

Fig.5 :Data mixed with redundancy bits

In the Hamming code, each r bit is the parity bit for one combination of data bits, is shown below:

R1 : bits 1,3,5,7,9,11

R2: bits 2,3,6,7,10,11

R4: bits 4,5,6,7

R8: bits 8,9,10,11

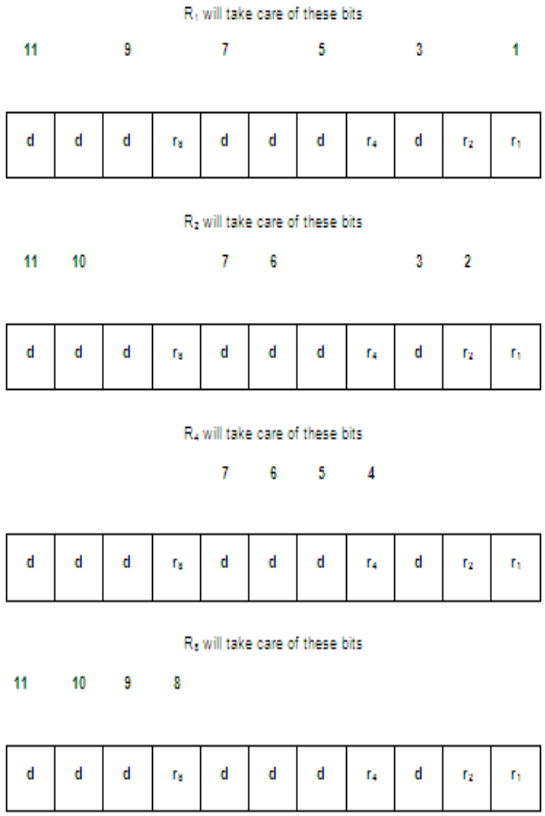

Fig 6 :Bits corresponding to the redundant bits

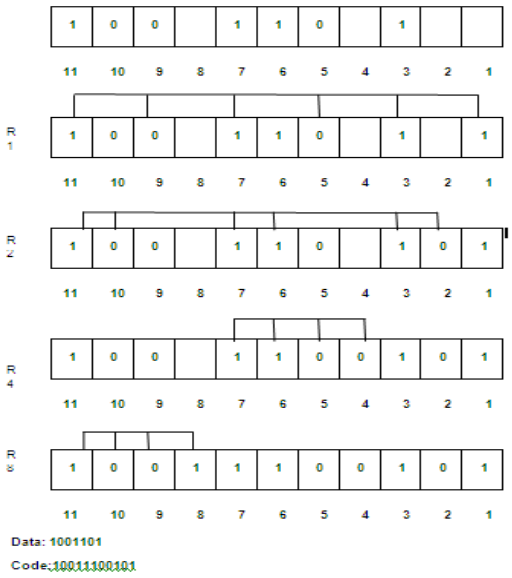

Fig 7 :Calulations of redundancy bits

Now imagine that by the time the above transmission is received, the number 7 bit has been changed from 1 to 0.The receiver takes the transmission and recalculates 4 new parity bits, using the same sets of bits used by the sender plus the relevant parity r bit for each set Then it assembles the new parity values into a binary number inn order of r position(r8 r4, r2 , r1). In our example, this step gives us the binary number 0111 (7 in decimal), which is the precise location of the bit in error.

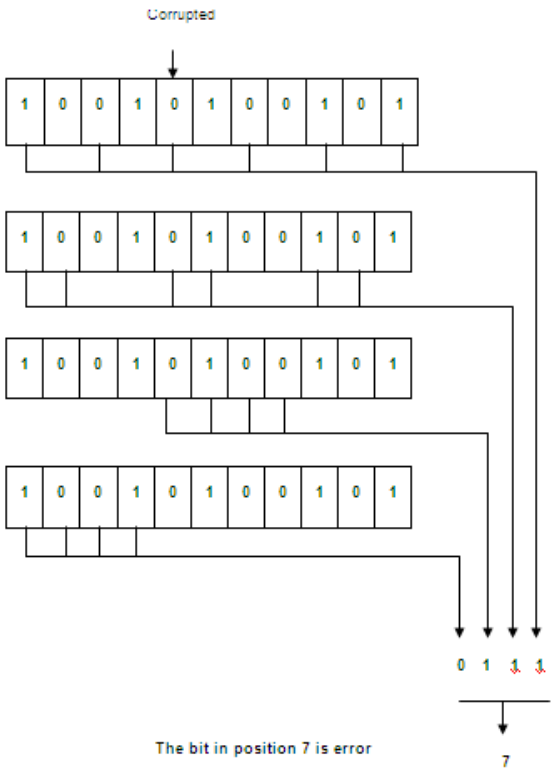

Fig 8:Calcutaion of the bit error position

Once the bit is identified, the receiver can reverse its value and correct the error. The beauty of the technique is that it can easily be implemented in hardware and the code is corrected before the receiver knows about it.

VI RESULTS AND DISCUSSIONS

The simulation output waveform views have been given.The design is simulated by using the simulation tool Xilinx ISE simulator 9.1,and waveforms obtained using Modelsim-Altera 6.3

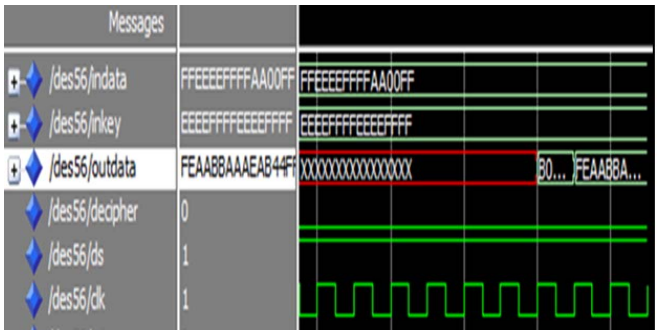

Fig 9 :Output waveform of encypted data.

Fig 9 shows the encryption of a plain text using des algorithm.The first two sections of the proposed system is same as for the existing system.The message is encrypted using a key and further send for compression.

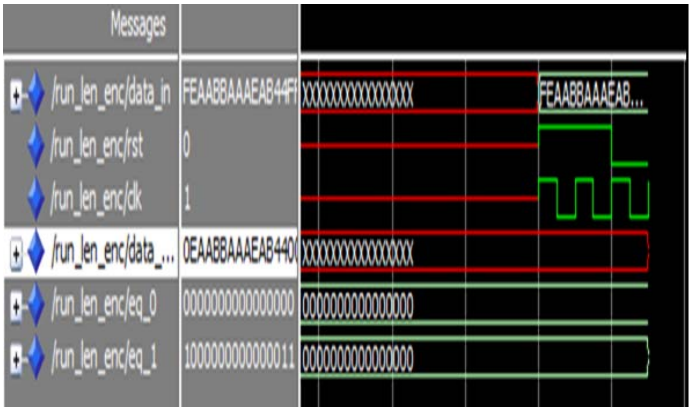

Fig 10:Output waveform of compressed data

Fig.10 shows the compression of the encrypted data.The cipher text obtained as a result of encryption is subjected to RLE where the input data is classified into a group of four from LSB bits,the they are checked for equivalence and when all bits in the group are same"1111" they are transmitted as "0000",and if the bits are "0000" they are transmitted as such.This is based on the logic that no transmission power is required for the transmission of zeroes.The postion where all ones and all zeroes are found are counted and registered using signals eq_0 and eq_1.These signals make the decompressing process possible.The detected positions enables the user to decompress the data without data losses.These are the results observed during the compression of the encrypted data.It could be found that where the bits are all ones are compressed and result is ones are replaced by zeroes.So during transmission power required can be reduced.

Input data :FEAABBAAAEAB44FF

Compressed Data:0EAABBAAAEAB4400

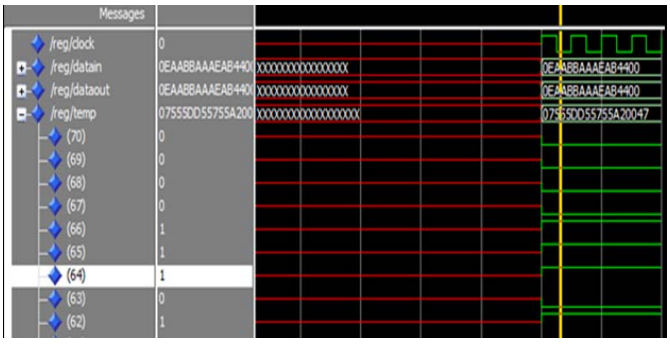

Fig 11: Case1: Output waveform the EDAC system if no error occurred

Fig.11 shows the performance of EDAC system when no error.The above shows a case when no error has occurred during transmission,so whatever data received by the EDAC it checks for the error,when no error has occurred during transmission then the same data is being transmitted.

Data in :0EAABBAAAEAB4400

Data out:0EAABBAAAEAB4400

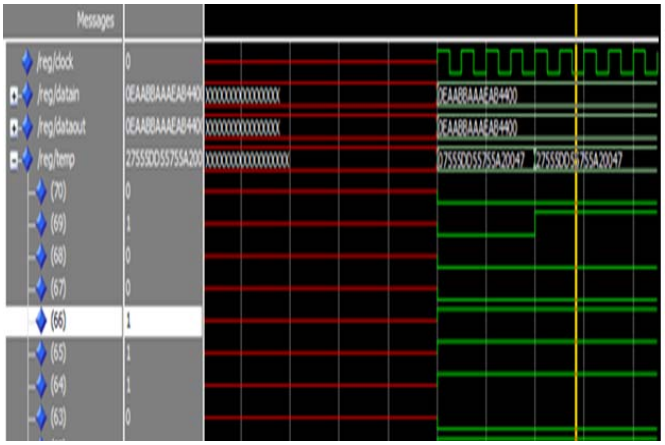

Fig 12: CaseII: Output waveform the EDAC system if one bit error occurred

Fig 12 shows the performance of EDAC system with one bit error This is a case when one bit error has occurred during transmission.Here the bit in the 69th position has altered and thus error has occurred.The EDAC system detects the error,and its position and corrects the error and the corrected data is given out.This was made possible using the Hamming codes,where the not just the number of errors are not just noted wherte the error bit position is calculated and hence the error is being corrected.With the help of syndrome generation the error in the bit position 69 was detected and then corrected to get the correct message.Such a system not just removes the errors whereas increase the overall efficiency of the system.The results obtained shows that the data out is same as that of data in even if there occurred a bit reversal in the transmitted data.

Bit reversal in the bit position no:69

Data in :0EAEAA00EEAA4000

Data out:0EAEAA00EEAA4000

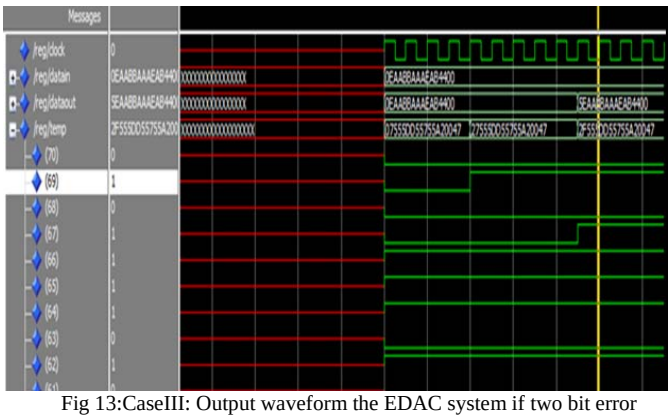

Fig 13:CaseIII: Output waveform the EDAC system if two bit error occurred

Fig.13 shows the performance of EDAC system with two bit errorThis is a case when more than one bit error has occurred during transmission.Here the bit in the 69thand 67thposition has altered and thus error has occurred.The EDAC system detects the error,Bit reversal in the bit position no:69 and 67 couldn't not be corrected.

Data in :0EAABBAAAEAB4400

Data out:5EAABBAAAEA**B4400**

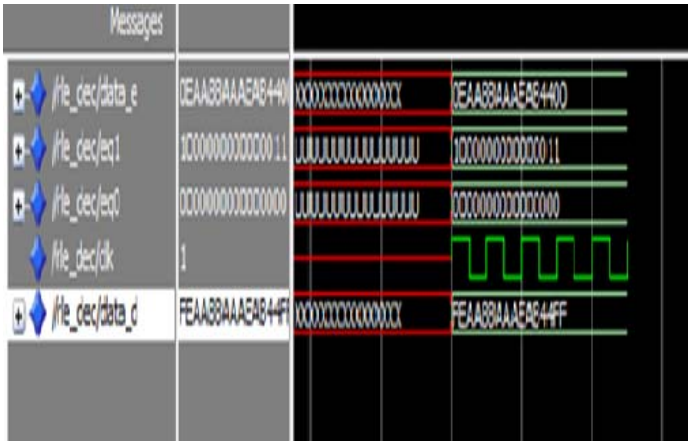

Fig 14 :Output waveform of Decompressed data

Fig 14 shows the decompresssion of corrected data.The positions of the zeroes,ones and compressed data are the input to this section,the ones are again replaced by zeroes and get back the cipher text.

Data in :0EAABBAAAEAB4400

Data out:FEAABBAAAEAB44FF

Fig 15 shows the decryption of the cipher text.The output of the decompressor is nothing but the cipher text,this cipher text is decrypted using the same key used for encryption to get back our original message or plain text.The overall working shows how the proposed system work as a whole even if an error encounters.

Data in :FEAABBAAAEAB44FF

Key :EEEEFFFFEEEEFFFF

Data out:FFEEEEFFFFAA00FF

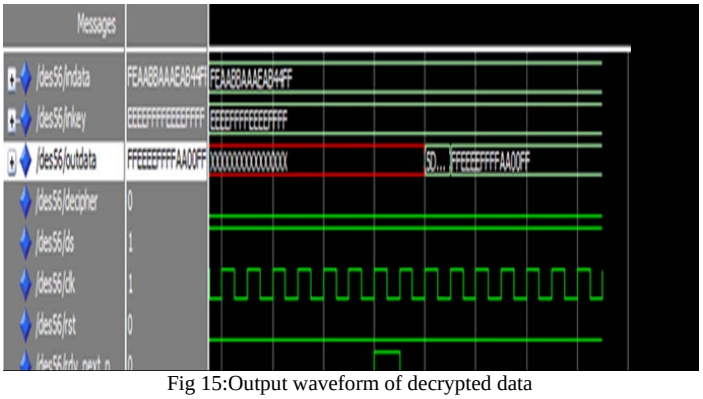

Fig 15:Output waveform of decrypted data

VII. CONCLUSION

In this work a error detecting and correcting circuit is included to the existing encryption and compression systems so as to improve the reliability of the system,there by reducing the bit error and thus help to maintain a better communication system where security,channel utilization is maintained along with error detection and correction mechanism.From the results it is clear that proposed work has error detection and correction capability as when compared to the traditional system.It is also shown that when an error occurs in the transmitted data,the data gets corrupted and cannot be decompressed as well as decrypted in the existing system,whereas with the help of the EDAC

system the errors are automatically detected and corrected.Thereby the overall efficiency and channel utilization is improved without compromising the security of the communication channel.

Further this work can be expanded to higher domains by using the same for AES,3DES algorithms and also the error correction capability can be increased from one bit to more.All these can considered as the future scope of this work.The whole system is coded using VHDL and and simulated using Xilinx ISE Suite 9.1 and output is obtained.

REFERENCES

[1] Aaron.A and Girod.B (2002),"Compression with side information using turbo codes," in Proc.IEEE Data Compress.Conf.,pp. 252–261.

[2] Abd El-Latif K.M.A.,H.F.A.Hamed, E.A.M.Hasaneen," FPGA Implementation of the Pipelined Data Encryption Standard (DES) Based on Variable Time Data Permutation", The Online Journal on Electronics and Electrical Engineering (OJEEE)

[3] Amandeep Singh, Manu Bansal, "FPGA Implementation of Optimized DES Encryption Algorithm on Spartan 3E", International journal of scientific & engineering research, volume 1, issue 1, october-2013issn 2229-5518

[4] Demijan Klinc, Carmit Hazay, Ashish Jagmohan, Hugo Krawczyk, and Tal Rabin ,"On Compression of Data Encrypted With Block Ciphers", ieee transactions on information theory, vol. 58, no. 11, november 2012

[5] ElGamal.T (1984), "A public-key cryptosystem and a signature scheme based on discrete logarithms," in Proc. CRYPTO, 1984, vol. LNCS-196, pp.10–18.

[6] Garcia-Frias.J,(Oct 2001), "Compression of correlated binary sources using turbo codes,"IEEE Commun.Lett.,vol. 5,no. 10,pp.417–419

[7] Gentry.C,(2009), "Fully homomorphic encryption using ideal lattices," in Proc. 41st Annu. ACM Symp. Theory Comput. pp. 169–178

[8] Jawahar Thakur, Nagesh Kumar,"DES,,AES:Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 2, December 2011)

[9] Johnson.M,Ishwar.P,Prabhakaran.V,Schonberg.D,andRamchandran. K(Oct 2004),"On compressing encrypted data," IEEE Trans. Signal Process,vol.52,no.10,pp.2992–3006,

[10] Liveris.A, Xiong.Z, and Georghiades.C,(Oct. 2002) "Compression of binary sources with side information at the decoder using LDPC codes," IEEE Commun.Lett., vol.6,no.10,pp.440–442,

[11] Nimmi Gupta "Implementation of Optimized DES Encryption Algorithm upto 4 Round on Spartan 3" International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2 , Issue 1