International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

MULTI BIT ERROR DETECTION & CORRECTION WITH AESENCRYPTION FOR SECURE COMMUNICATION

[1]A SRINIVASA REDDY, [2]A.ASWANILALITHA

[1]M.Tech Scholar, Dept of ECE, Chalapathi Institute of Technology, A.R Nagar, Mothadaka, Guntur-522016: Andhra Pradesh

[2]Assistant Professor, Dept of ECE, Chalapathi Institute of Technology, A.R Nagar, Mothadaka, Guntur-522016: Andhra Pradesh

ABSTRACT: This paper presents new enhanced method to encode the message, detects the error, and corrects the message in the communication processes. These methods have been developed based on Hamming based multi-bit error detection and correction technique with encryption. The key point for the implementation of error-free communication is the encoding of the information to be transmitted in such a way that some extent of redundancy is included in the encoded data, and a method for efficient decoding at the receiver is available. These two requirements have been achieved in the new method in an efficient and simple way Temporary errors which are classified under soft errors are created because of fluctuations in the voltage or external radiations. These errors are very common and obvious in memories. In this paper, Diagonal Hamming based multi-bit error detection and correction technique is proposed to identify errors 1 bit error for one row. The new methods are demonstrated using some examples, and have given good result.

KEY WORDS: Multi Bit Error Detection & Correction, Diagonal Hamming, error-free communication.

I.INTRODUCTION

A central problem of coding theory is reliable communication over an unreliable channel. All solutions to this problem, in some form or another, depend on the basic idea of encoding messages with some redundancy, allowing the receiver to detect and correct whatever errors may arise during transmission through the channel. The main goal is to minimize the amount of redundancy while maximizing the quantity of errors that can be corrected. Networks and other communication systems must be able to transfer data from the source to the receiver with complete accuracy. A system that cannot guarantee that the data received by one device are identical to the data transmitted by another device is essentially useless.

The theory of error correction is concerned with sending reliably information over a noisy channel that introduces errors into the transmitted data. The goal of this research is to design coding schemes which are capable of detecting and correcting such errors. The setting is usually modeled as follows: a transmitter starts with some message, which is represented as a string of symbols over some alphabet. The transmitter encodes the message into a longer string over the same alphabet, and transmits the block of data over a channel. The channel introduces errors (or noise) by changing some of the symbols of the transmitted block, and then delivers the corrupted block to the receiver. Finally, the receiver attempts to decode the block, hopefully to the intended message.

Whenever the transmitter wants to transmit a new message, the process is repeated. Two factors are of special interest in this setting. The first is the information rate, which is the ratio of the message length to the encoded block length. This is a measure of how much "actual message data" is carried by each transmitted symbol. The second is the error rate, which is the ratio of the number of errors to the block length. This is a measure

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

of how "noisy" the channel is, i.e. how much data it corrupts.

Of course, we desire coding schemes that tolerate high error rates while simultaneously having large information rates. In practice, smaller alphabets are desirable too, as most digital communication devices are, at their lowest levels, capable of interpreting only binary digits (bits). BINARY information is stored in a storage space called memory. This binary data is stored within metal-oxide semiconductor memory cells on a silicon integrated circuit memory chip.

Memory cell is a combination of transistors and capacitors where capacitor charging is considered as 1 and discharging considered as 0 and this can store only one bit. Errors which are temporary or permanent are created in the memory cells and need to be eliminated. Single bit error correction is most commonly used technique which is capable of correcting upto one bit. Since technology is increasing rapidly, there are more probabilities of getting multiple errors. Use of Diagonal Hamming method leads to efficient correction of errors in the memories. Memory was divided as SRAM, DRAM, ROM, PROM, EPROM, EEPROM and flash memory.

Main advantages of semiconductor memory are easy to use, less in cost, and have high bits per square micrometers. Temporary errors are called transient errors which are caused because of fluctuations in potential level. Permanent errors are caused because of defects during manufacturing process or large amount of radiations.

II. TYPES OF ERRORS

Whenever an electromagnetic signal flows from one point to another, it is subjected to unpredictable interference from heat, magnetism and other forms of electricity. This interference can change the shape or timing of the signal. The error can be:

• **Single-Bit Error**: The term single bit error means that only one bit of a given data unit (such as a byte, character, data unit, or packet) is changed from 1to 0 or from 0 to 1.

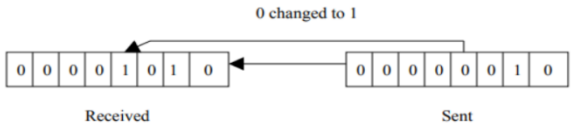

Fig: 1: SINGLE BIT ERROR

Single- bit error are the least likely type of error in serial data transmission. For a singlebit error to occur the noise must have duration of only 1 microsecond, which is very rare; noise normally lasts much longer than this. However, a single –bit error can happen if we are sending data using parallel transmission. For example, if eight wires are used to send all of the eight bits of a byte at the same time and one of the wires is noisy, one bit can be corrupted in each byte.

Burst Error: The term burst error means that two or more bits the data unit has changed from 1 to 0 or from 0 to 1. A burst error doesn't necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some it's in between may not have been corrupted.

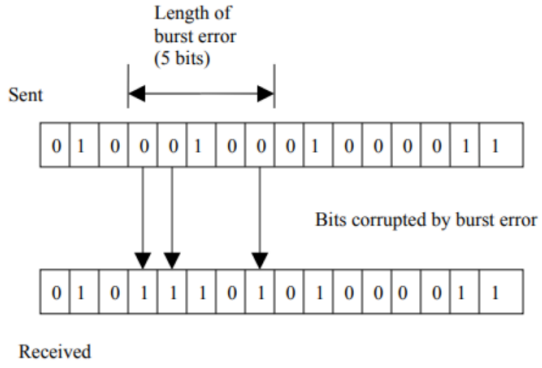

Fig: 2: BURST ERROR

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Burst error is most likely to happen in a serial transmission. The duration of noise is normally longer than the duration of a bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at 1 Kbps, a noise of 1/100 seconds can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits.

III. ADVANCED ENCRYPTION STANDARD (AES) CRYPTOGRAPHY

The Advanced Encryption standard is a 128-bit block cipher that has beenwidely used since its acceptance in 2001. The design of AES was intended to be a more secure replacement of DES (Data Encryption Standard). Many efficient hardware and software designs have been documented, taking into consideration various tradeoffs of speed and area resources. The following sections will provide a general functional description of AES with an increased focus on the hardware design of AES components. High speed hardware datapathsthat will be relevant in understanding the GCM datapath will be presented toward the end of this section.

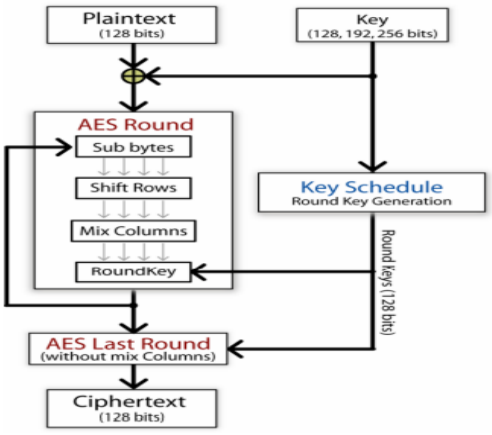

Fig. 3: AES ROUND STRUCTURE

The 128 bit plain text input is mapped into a state array which is a 4x4 block of 8 bit words that is manipulated in each round. For the following sections the state array block will be used to describe the different round operationsso it isimportant to understand how the input is transformed into the state array.

Byte Substitution (Subbytes): The subbytes operation uses multiple substitution box components (Sbox) each of which performs an 8-bit substitution. Each 8-bit word of data in the state array, is substituted using the Sbox. This results in 16 Sbox components used for each round block, and is the most hardware area consuming part of an AES round. The Sbox computation is essentially a multiplicative inverse in GF (28) followed by an affine transformation which is a linear mapping from one vector space to another. A lookup table of 28 values can be used to implement the Sbox component, but it can also be mathematically computed using logic gates.

Shift Rows: The Shift Rows operations consists of cyclically moving elements around in all but the first row of the 128 bit input block. The rows are left shifted by 1, 2, and 3 times respectively for rows 2, 3 and 4. The following mapping illustrates this process. In hardware no logic is required for this step and simple wire connections are used for this step to route the input to the output.

Mix Columns: The mix columns operation consists of a multiplication and reduction operation over GF(28). Each column of the state array is multiplied by the polynomial 3x 3+ x 2+x+2 and reduced modulo the field generating polynomial x 4+1. This operation is generally optimized into a single matrix vector product. The four column blocks are

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

used as the vectors, while a constant 4x4 matrix is used that combines the modulo operation. The result vector is stored in the next state array at the same location as the original column vector.

All elements are 8 bits in width and the multiplication and addition operations are performed over GF(28) Since the elements of the matrix are of low degree the multiplications are simplified. A multiplication with 2 in GF(28) consists of a shift operation along with a modulo reduction.if an overflow occurs. This operation can be reused with mul- tiplying by 3, but an extra addition is required since $3 \cdot ai = (2 \cdot ai) \oplus ai$.

Key Schedule Round keys are XORed at the end of every round and are generated using a Key Schedule. These keys can be precomputed or generated at each round. The Sbox components used in the subbytes section are also used here for the round key generation. For each inputted key length, the method of generating keys is slightly different, but they contain similar logic components. The 128 bit key has an Sbox operation done on the last column of the cipher key state array after the column bytes are rotated.

This is followed by a rcon value XOR addition. The rcon value is generated based on the exponentiation formula rcon(i) = x 254+i mod x 8 + x 4 + x 3 + x + 1 performed over GF(28). These values are usually precomputed and once the rcon value is added there is an XOR chain on the columns of the state array that creates the next 128 bit round key .this process is repeated by using the round key as a cipher for generating the next 128 bits of key material. The rcon i value starts at 1 and increments for each round key.

The 192 bit key schedule is similarly defined but the XOR chain is extended for another 2 columns to achieve a full 192 bits of key material. Each round unit of AES, however, only uses 128 bits of key material at a time, so the remaining bits are carried over for the next round. The six column vectors of the key state array are condensed here and shown as {A0, A1, · · · , A5}. The 256 bit key schedule has an additional Sbox computation involved in gen- erating key material. The first 128 bits of key material is generated as shown in Figure 3. Forthe next 128 bits of key material, an Sbox computation is performed on the fourth column and this follows another chain of XOR statements. Note that the rcon operation is not performed here with the Sbox.

IV. ERROR DETECTION & CORRECTION WITH AES PROCEDURE

Encoding In encoding technique message bits are given as input to the Diagonal Hamming encoder and the Hamming bits are calculated. This process taken the 32-bit message. Message and Hamming bits (32+24 bits) are saved in the memory after the encoding technique. Based upon the message input need to combine the message bits and hamming bits based on the diagonal Hamming bits process of division of message bits.

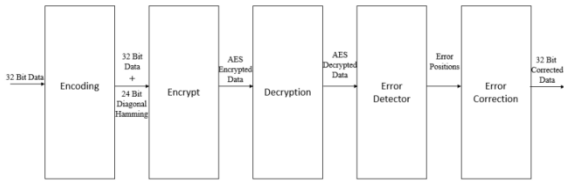

Fig. 4:PROPOSED ARCHITECTURE FOR ERROR DETECTION AND CORRECTION WITH SECURED DATA

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Encryption From the hamming encoded data, encrypting for the security purpose with AES technique, this technique we need the 128 bit data but our message data was 64 bit for that reason taken the same message multiple times or can go for adding remaining bits as empty data as well. In this taken as duplicating the same message and made it like 128-bit data. After this encrypt the data with the AEC technique of SubBytes, ShiftRows, MixColumns, Add Round Key along with our key. Completion of encryption will transfer the data.

Decryption After receiving of the data need to decrypt the data at the transmitter end with same AES decryption technique with our key then the message will be converted as the hamming converted code with noise.

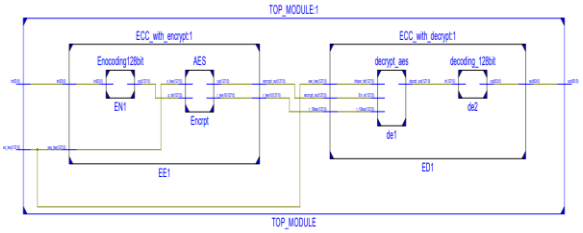

Fig. 5: BLOCK DIAGRAM OF 32-BIT DIAGONAL HAMMING CODE WITH CRYPTOGRAPHY

Decoding Decrypted data will be decoded with hamming technique if the noise came in data.If all the syndrome bits are equal to zero, then it represents that the message bits are not corrupted and if anyone of the syndrome bits in non-zero, then it represents the message bit(s) are corrupted. These corrupted bits need correction so, the message bits are sent to the error correction part. In the correcting of error part, the location of error is identified by doing the following calculations: Suppose if the error is in the third row of the message organization, then the error position is calculated as: (S3[3] ∗ (22)) + (S3[2] ∗ (21)) + (S3[0] ∗ (20)); (49) After the position is calculated, the error corrector negates the bit corresponding to that position to correct the data bit. This process is done till all the corrupted bits are corrected.

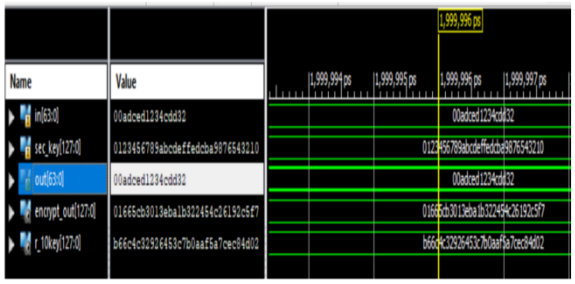

Fig. 6: SIMULATION OF 32-BIT DIAGONAL HAMMING CODE WITH CRYPTOGRAPHY

V. CONCLUSION

ECC has many applications in communication industry, so if it is efficient in terms of its design metrics then it will help in improving efficiency of data transmission. So, in this project concentrated on design the 32-Bit Error detection and Correction with encryption. This approach will prove the secured communication without loss of data with less complexity.

VI.REFERENCES

[1]. Boccini, G. Security in Automotive Microcontrollers of Next Generation. Ph.D. Thesis, Università di Pisa, Pisa, Italy, 2014.
[2]. Mundhenk, P.; Paverd, A.; Mrowca, A.; Steinhorst, S.; Lukasiewycz, M.; Fahmy, S.; Chakraborty, S. Security in Automotive Networks: Lightweight Authentication and Authorization. ACM Trans. Des. Autom. Electron. Syst. 2017, 22, 1–27. [CrossRef]
[3]. Ni, X.; Shi, W.; Foo, V.F.S. AES Security Protocol Implementation for Automobile Remote Keyless System. In Proceedings of the 2007 IEEE 65th Vehicular Technology Conference—

VTC2007-Spring, Dublin, Ireland, 22–25 April 2007; pp. 2526–2529. [CrossRef]
[4]. Lv, X.; Xu, L. AES encryption algorithm keyless entry system. In Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, 21–23 April 2012; pp. 3090–3093.
[5]. Lugo-Meneses, C.A.; Peralta-Reynoso, D. Secure AES Frame Encryption for CAN FD. 2019. Available online: https://rei.iteso.mx/handle/11117/5973 (accessed on 21 August 2021).
[6]. Henniger, O.; Ruddle, A.; Seudié, H.; Weyl, B.; Wolf, M.; Wollinger, T. Securing Vehicular On-Board IT Systems: The EVITA Project. 2009. Available online: https://www.evita-project.org/Publications/HRSW09.pdf (accessed on 21 August 2021).
[7]. Jattala, I.; Durrani, S.; Farooqi, J.; Junjua, G.; Shafique, A.; Hussian, F.; Mahmood, H.; Ikram, N. Secure automotive telematics system (SATS). In Proceedings of the Eighth International Conference on Digital Information Management (ICDIM 2013), Islamabad, Pakistan, 10–12 September 2013; pp. 262–267. [CrossRef]
[8]. Cassettari, R.; Fanucci, L.; Boccini, G. A new hardware implementation of the advanced encryption standard algorithm for automotive applications. In Proceedings of the 2014 10th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), Grenoble, France, 30 June–3 July 2014; pp. 1–4.
[9]. Traoré, P.S.; Asfour, A.; Yonnet, J.P. Noise analysis of a high sensitivity GMI sensor based on a Field-Programmable-Gate-Array. Sens. Actuators A Phys. 2021, 331, 112972. [CrossRef]
[10]. Toan, N.; Tung, D.; So, J.; Lee, J.G. Immunity Characterization of FPGA I/Os for Fault-Tolerant Circuit Designs against EMI. Adv. Electr. Comput. Eng. 2019, 19, 37–44. [CrossRef]
[11]. Benfica, J.; Green, B.; Porcher, B.C.; Poehls, L.B.; Vargas, F.; Medina, N.H.; Added, N.; de Aguiar, V.A.P.; Macchione, E.L.A.; Aguirre, F.; et al. Analysis of FPGA SEU sensitivity to combined effects of conducted EMI and TID. In Proceedings of the 2016 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), Shenzhen, China, 17–21 May 2016; Volume 1, pp. 887–889. [CrossRef]
[12]. Shum, W. Glitch Reduction and CAD Algorithm Noise in FPGAs. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2011.