Volume 8, Issue 5
October 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Development of Hardware-Based Advanced Encryption Standard with Error Correction Capabilities

[1]Kinny Garg and [2]J. S. Sohal

[1]Ph.D. Research Scholar, I. K. Gujral Punjab Technical University, Jalandhar, India.

[2]Director, Ludhiana College of Engineering and Technology, Katani Kalan, Ludhiana, India.

Email: [1]kinnygarg25@gmail.com, [2]jssohal2001@yahoo.co.in

ABSTRACT

The importance of cryptography is continuously increasing since the amount of sensitive data being transmitted over communication channels is growing at an unprecedented pace. Data security and correctness are some of the critical objectives for data communications in an environment of increasing data thefts and malicious attacks. Thus, a significant area of concern for data communications today is the lack of robust data encryption equipment equipped with correction capabilities. Advanced Encryption Standard (AES), which is one of the most robust encryption standards to date, has no embedded error correction capabilities. In this paper, we augment AES with error correction capabilities using Hamming Code. Our algorithm, which has been developed in Verlilog HDL, has the ability to auto-correct single-bit errors introduced in the encrypted data transmitted over a noisy channel.

Keywords: *Advanced Encryption Standard, Cryptography, Error Correction Code, Hamming Code*

1. INTRODUCTION

The use of wireless data communications is growing day by day. The need to keep the data secure and private requires it to be protected from attacks as well as unwanted noise in the communication channel. So, cryptography and error-correction coding are two essential ingredients of secure communications. There are many commonly used algorithms for encryption, but we have focused on the AES standard because it has "Confusion and Diffusion" property for the secure channel data. It also protects from the malicious attacks. Confusion refers to the property that each ciphertext character depends on many parts of the key, and diffusion refers to the property that several ciphertext characters should change when we change a plaintext character [1]. To achieve this objective, AES would be beneficial because of its key size, security, fast in both hardware and software, and its ubiquity. The three AES versions are based on keys of three different lengths (128, 192 and 256 bits). The default key size is 128 bits [1]. This paper proposes a novel way to secure the data for transmission over a channel from malicious attacks by implementing the robust encryption standard, viz., the AES standard [1].

In addition to the data security concerns, when the channel of data communication is noisy, the data is likely to get corrupted. Error correction codes can be used to augment self-data correction functionality to transmitted data at the expense of increased requirement for bandwidth. In this paper, we add a layer of Hamming code onto existing AES encryption code for adding error-correction capability to the AES standard [2].

2. LITERATURE REVIEW

A vast number of studies have been carried out in the field of cryptography. Until recently, the Data Encryption Standard (DES) was considered to be one of the most robust encryption standards and was probably the most used encryption standard in practice world-wide. However, with the advent of Advanced Encryption Standard (AES), the DES is being replaced with AES.

The existing literature on the subject addresses the implementation of AES through two means, viz., hardware and software. The software implementation of the AES algorithm is an easy but slower and less secure process [3]. So, when high security and high throughput are the objectives, hardware implementations serve better.

Attempts have been made by research scholars and scientists to implement the AES algorithm in numerous ways, each differing in degree of fastness and silicon area optimization. Some researchers have presented ideas for detection of faults, which an attacker may inject maliciously while the data is encrypted or decrypted to find the secret key. Some researchers have made attempts even to correct these random faults that get introduced during the encryption/decryption process [4]. However, none of the existing work on this subject in the literature, to the best of our knowledge, addresses the self-error-correction capability of the encrypted message for data communications. These capabilities, of course, come at the cost of increased hardware requirements.

3. DESIGN

In this paper, we aim to arrive at a hardware-based optimized version of AES with limited error correction capabilities. The block diagram given in Figure 1 depicts the high-level design of our study.

Volume 8, Issue 5
October 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

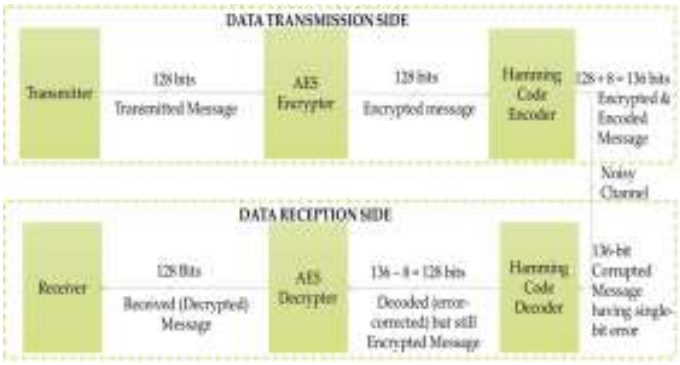

Figure 1: Block Diagram of Required Work

A communication system has two major components – a transmitter to send the data and a receiver to receive the transmitted data. The function of the transmitter is to convert the message signal into a form suitable for transmission across the channel or medium of communication. In the block diagram, on the transmission side, there are two major blocks, i.e., AES encrypter and Hamming code encoder. Our first objective is to secure data from malicious attacks using the AES standard. The function of the AES Encrypter block is to preserve the 128-bit data by encrypting it using a 128-bit key.

Further, executions can be done by 192 bits and 256 bits. For each of the sizes specified, AES utilizes a distinct number of rounds. When using a 128-bit key, 10 rounds of execution must be used. Using a 192-bit key, use 12 rounds of implementation and perform 14 rounds when using a 256-bit key [5].

In the block diagram shown in Figure 1, plaintext and key (in hexadecimal form) are given as inputs to the AES encrypter. Th AES standard based on a 128-bit key has a total of 10 rounds to perform to get the output of the AES encrypter. The round feature consists of four distinct byte-oriented transitions:

- Replacement byte using Substitution Box(S-Box) (SubBytes),
- Different offset rows of the state arrays (ShiftRows),
- Mix the data in each state array column (MixColumns)
- Add the round key (Add round key) to the state.

In all the 10 rounds, above four transformations are performed identically till the final round [6]. Mixing the data within each column (MixColumns) does not happen in the final round. We get the output of 128-bit (Hexadecimal). All vectors in input and output are in hexadecimal notations. The left character of each pair provides the 4-bit group with the higher numbered bits and the right character of each pair produces the lowered numbered bits. For all bytes within these test vectors, the array index begins at zero and steps-up from left to right [7]. Now the output of the AES encrypter, i.e., ciphertext will be given to the input of the hamming code encoder, which is used according to their advantages. As a basic principle of the coding of channels, hamming code is a linear error-correcting code. It enables the correction of transmission errors without the need for a correct transmission feedback channel. Hamming code called after the inventor, Richard Hamming [8].

By using this code, two concurrent bit errors can be detected, and a single bit error can be corrected. The input of the Hamming code is also 128-bit long. There are two ways to generate redundancy bits (even parity and odd parity). The number of bits of redundancy relies on the size of the data bits of information. The code (n, k, t) relates to an ' n'-bit codeword with ' k ' information bits (where n > k) and ' r ' (= n – k) parts of error control called ' redundant ' or ' redundancy ' bits with code capable of correcting ' t ' bits in error (i.e., ' t ' corrupted bits). If the total number of bits in a transmittable unit (i.e., codeword) is ' n ' (= k + r), the number of bits in a transmittable unit must be at least ' n+1 ' (= k + r + 1). Of these, one state does not mean error and n ' state indicates where an error occurs in each of the ' n ' positions. So 'n+1' states must be discoverable by 'r' bits; and 'r' bits can indicate 2^r different states [8]. Therefore, 2^r must be equal to or greater than 'n + 1':

$2^r \geq n + 1$, or

$2^r \geq k + r + 1$

We can determine the value of ' r ' by replacing the value of ' k ' (the initial length of the data to be transferred). Here, we have '128,' is the value of "k" and the value of 'r' should be the smallest that can satisfy this constraint is '8'. $2^8 \geq 128 + 8 + 1$. Now, we have got 8 redundancy bits. These redundancy bits help us to detect the data. The data bits along with the redundancy bits to become a codeword [9]

At the receiver side, again, we have two major blocks, i.e., Hamming code decoder and AES decrypter. The receiver side receives 136-bit data, 128-bit encrypted data, and 8 redundant bits. The receiver gets 136-bit encrypted data at the destination and checks for any possible errors. If any single-bit error occurs, the receiver will find and correct the location of the error. Hamming decoder detects and corrects the error by XORing data by a NOT gate. The ciphertext is coming from the previous stage, so AES decrypter is used to get the plaintext of 128 bits. The decryption method is the method of obtaining the initial encrypted information. This method is based on the key the sender got. AES decryption processes are similar to the reverse order encryption process [9], and both sender and receiver have the same key for encrypting and decrypting data. The final round of a decryption phase comprises of three stages like InvShiftRows, InvSubBytes, and AddRoundKey [10].

4. RESULTS

We use the Verilog Hardware Description Language (HDL) to implement our algorithm depicted in the block diagram given in Figure 1.

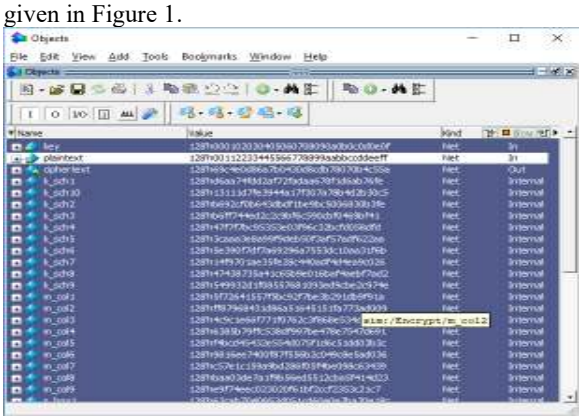

Figure 2: Output of the Encrypter

Volume 8, Issue 5
October 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Figure 2 shows the output of the encrypter (as mentioned in the block diagram) in which plaintext and key are inputs of the encrypter. We get the ciphertext as an output of the encrypter.

PLAINTEXT(Input):00112233445566778899aabbccddeeff
KEY(Input):000102030405060708090a0b0c0d0e0f

The plaintext is in blocks of 128 bits, and also the key which we have used is 128-bit long. After the AES algorithm, the number of rounds to be performed during the execution of the algorithm relies on the size of the key of the AES algorithm. Here the number of rounds will be 10. So, after performing the 4 steps of the AES algorithm, we get the ciphertext as an output.

CIPHERTEXT(Output):69c4e0d86a7b0430d8cdb78070b4c55a

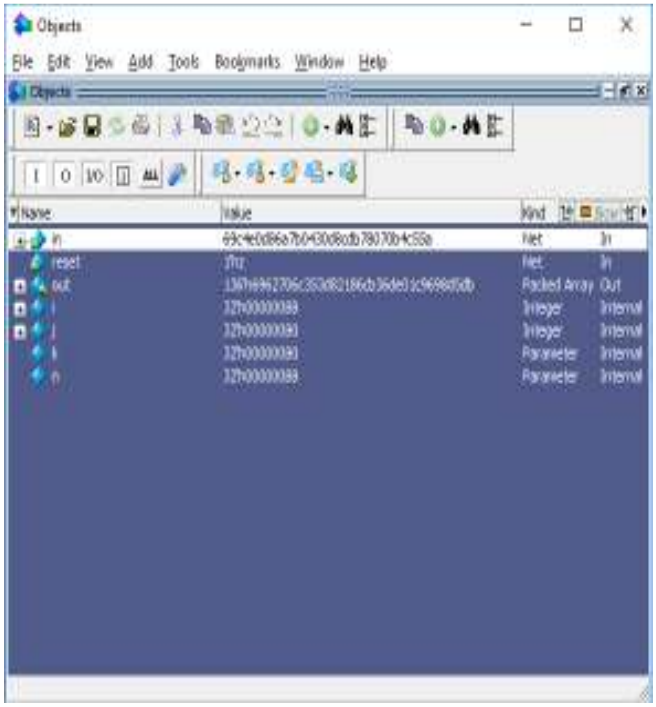

Figure 3: Output of the Encoder

Figure 3 shows the output of the encoder. IN represents, the input of 128 bit and OUT represents the output of the encoder of 136 bits with 8 redundancy bits.

IN(Input):69c4e0d86a7b0430d8cdb78070b4c55a
OUT(Output):6962706c353d82186cb36de01c9698d5db

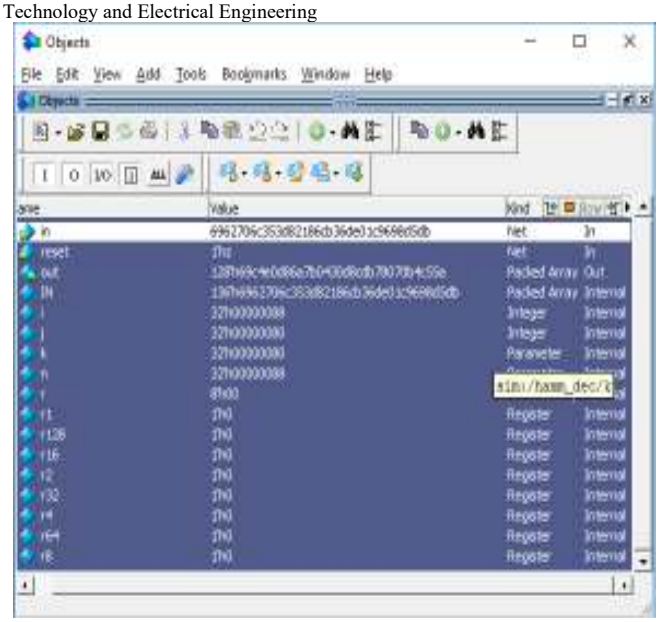

Figure 4: Output of the Decoder

Figure 4 shows the output of the encoder. IN represents, the input of 136 bits and OUT represents, the output of the encoder of 128 bits.

IN(Input):6962706c353d82186cb36de01c9698d5db
OUT(Output):69c4e0d86a7b0430d8cdb78070b4c55a

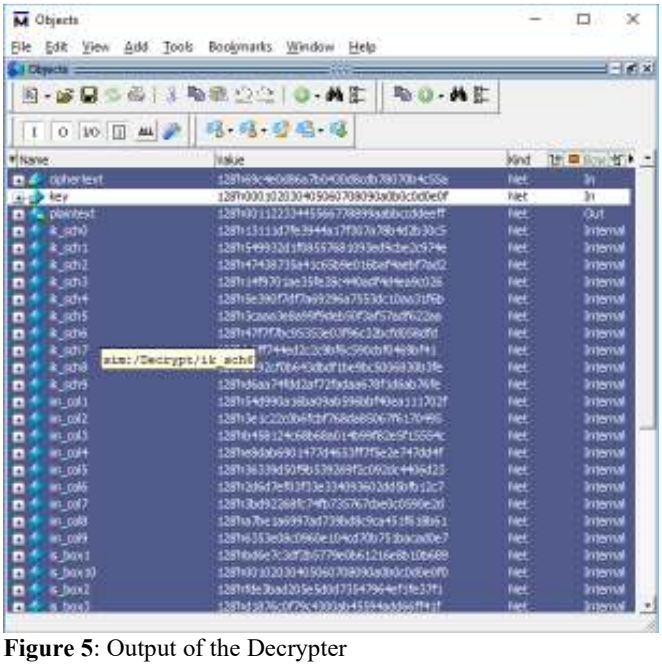

Figure 5: Output of the Decrypter

Figure 5 shows the output of the decrypter in which ciphertext is converted into plaintext after performing the same (10) rounds as in encryption. We get the plaintext text as an output of the decrypter.

CIPHERTEXT(Input):69c4e0d86a7b0430d8cdb78070b4c55a
KEY(Input): 00112233445566778899aabbccddeeff
PLAINTEXT(Output):00112233445566778899aabbccddeeff

Volume 8, Issue 5
October 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

5. CONCLUSION

The use of the internet and the network is quickly growing. Quite often, information that needs to be transmitted over a network is vulnerable to malicious attacks and should, therefore, be protected from intruders. Encryption and decryption algorithms play an important role in preventing unauthorized access to data. In this paper, we have developed the 128-bit key-based AES with single-bit error correction capabilities.

AES supports three possible key lengths to enable users to make a tradeoff between speed and safety. If we increase the key length, then the execution time of encryption and decryption also increases. Future research work can focus on the AES standard with a key of 256-bit length with different error correction capabilities. Future studies can also focus on multi-bit error correction capabilities.

ACKNOWLEDGMENT

We would like to acknowledge the support offered by I.K. Gujral Punjab Technical University, Jalandhar, Punjab, India by means of providing access to the latest journals and making available the resources.

REFERENCES

[1] National Inst. Of Standards and Technology, "Federal Information Processing Standard Publication 197, the Advanced Encryption Standard (AES)," Nov. 2001.
[2] Kinny Garg, J.S. Sohal "FPGA-based Implementation of Galois Field Arithmetic", International Journal of Modern Electronics and Communication Engineering Volume No.7, Issue No.2, pp.6-9, 2019.
[3] J. Daemen and V. Rijmen," AES Proposal: Rijndael," AES Algorithm Submission, Sept. 1999.
[4] William Stallings, Cryptography and Network Security, Principles and Practices, 4th ed. Pearson Education, pp. 134-161, 2006.
[5] Wayne Wolf, "FPGA-Based System Design, Pearson Education, pp. 17-37.
[6] Xinmiao Zhang and Keshab K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm," IEEE Circuits and Systems Magazine, vol. 2, no. 4, pp. 24–46, 2003.
[7] Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Private Communication in a Public World, 2nd ed. Pearson Education, pp. 41-114, 2006.
[8] R. W. Hamming, Error-detecting and error-correcting codes, Bell System Technical Journal, 29(1950), 147–160.
[9] J. Wolf, B. Elspas, Error-locating codes a new concept in error control, IEEE Transactions on Infomation Theory,9(2) (1963), 113–117.
[10] Chih-Chung Lu and Shau-Yin Tseng, "Integrated Design of AES (Advanced Encryption Standard) Encrypter and Decrypter," Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, and Processors, (ASAP'02), pp. 277-285, 2002.

AUTHOR PROFILES

Kinny Garg is pursuing Ph.D. from I.K.G. Punjab Technical University, Jalandhar, India. She has an experience of more than 5 years in teaching. She has received M.E. degree in Electronics and Communication Engineering from Thapar University, Patiala in 2009 and B.Tech. degree in Electronics and Communication Engineering from Giani Zail Singh College of Engineering & Technology, Bathinda in 2007. Her research interests are in the area of networking, wireless communication, and cryptography.

Dr. J.S. Sohal is the Director of Ludhiana College of Engineering and Technology, Katani Kalan, Ludhiana. He has a professional experience of more than 35 years spanning across teaching, research, and administration. He has served as the Dean FET, Prof & Head DET, and DSW at GNDU, Amritsar. He has also served as a Professor and the Head of the Department of Computer Science and Electrical Engineering, PAU, Ludhiana. He has been a member of the Academic Council of BOS in CSE & IT at PTU, Jalandhar. He has guided 7 Ph.D. students and several M.Tech. students.