AES encryption and decryption standards

1Sistla Vasundhara Devi and 2Harika Devi Kotha

12 Department of Electronics and Communciation Engineering, Faculty of Science and Technology
ICFAI Foundation for Higher Education, Hyderabad, India
Corresponding Author Email: svdevi97@gmail.com.

Abstract: Over the last years there has been massive changes that lead to the growth of information in technology, that brought significant changes in the part of cryptography and its applications like confidentiality of data and many such. Privacy and secrecy is what everyone desires for their data or accounts. Encryption is one such method to achieve it. Advanced Encryption Standard (AES), can be used to protect the information. The primary preferred standpoint of AES is, it very well may be can be reproduced or worked with unadulterated equipment. In this paper, Xilinx 9.2i is utilized for recreation and improvement of VHDL code. Integrating and execution of the code is completed on Xilinx - Project Navigator ISE 9.2i suite. Xilinx XC3S500 gadget of Spartan Family is utilized for equipment usage. This undertaking proposes a strategy to coordinate the AES encrypter and the AES decrypter.

Index Terms: component, formatting, style, styling, insert, VHDL, AES, DES.

1. Introduction
The Advanced Encryption Standard additionally called as AES is a calculation intended for the improvements that are the then earnest needs to make information increasingly secure. Thus the U.S Government held challenges in 1997 to the encryption and unscrambling norms improvement, in which the AES is the victor. The past form called the Data Encryption Standard was observed to be frail in doing this encryption business because of its little key size and mechanical headways in processor control. The following advancement was distinguished in October 2000 which is a somewhat adjusted variant of the Rijndael (this name depends on its two Belgian innovators Joan Daemen and Vincent Rijmen) known as square figure AES underpins square sizes of 128-bits and key sizes of 128, 192, 256-bits. Though unique Rijndael underpins key and square sizes of least 128 to 256-bits, it very well may be any numerous of 32. To scramble messages longer than the square size, a method of activity is picked which is extended at alternate pieces of this article.

1.1 Cryptography: ensures the confidentiality of the information by converting original data into cipher text. A cryptographic system works by transforming plaintext into a cipher text, using key. However, the applications of cryptography go far beyond simple confidentiality [1]. With the use of cryptography one can assure the authenticity and integrity of the information as well as the users [2].

1.2 Overview of Encryption and Decryption
There any many network examples which are the open networks like bank account details, credit card transactions, tax details etc., which need to be more secured, for this the encryption algorithm is very much useful to make sensitive information of the user safe to transmit over any open network by scrambling into a form that cant be understandable by any hacker. To do this we use mathematical formulas, hence called encryption algorithm, this transforms message bits into an unintelligible form. which the intended recipient having the decryption algorithm can only decrypt to see the original message. Hence using this algorithm the above stated examples are now used more safe and secure manner.
1.3 Cipher Key
A cipher calculation is a numerical equation planned explicitly to cloud the esteem and substance of information. Most significant cipher calculations utilize a key as a major aspect of the recipe. This key is utilized to scramble the information, and either that key or a corresponding key is expected to unscramble the information back to a helpful structure. Many cipher calculations increment their assurance by expanding the span of the keys they use. Be that as it may, the bigger the key, the all the more processing time is expected to encode and decode information. So it is vital to pick a cipher calculation that strikes a harmony between your assurance needs and the computational expense of securing the information. Essentially, encryption calculations fall into two classes relying upon key utilization. Those calculations which contain two keys one open for encryption and other private key are classified under hilter kilter. Those calculations by and large for which utilizes one open key are named as symmetric. In a symmetric or private key calculation, as a rule, we utilize just a single key for encryption and decoding of the information.

1.4 Stream Cipher and Block Cipher
Stream figures forms substantial squares of information, though stream figures regularly work on littler units of plaintext, for the most part bits. Stream figures are quicker than square ciphers. In stream figure Encryption is finished by joining the key stream with the plaintext, by and large bitwise XOR task is performed. Synchronous stream figure is where key age is autonomous of info text. (Stinson, 2002). In square figure, square size can be 64 or 128, 192 or 256 bits.

2. AES evaluation
The previous version of encryption i.e DES is now not considered that secured for many different applications due to the reason of its key size being 56 bit which is too small, and hence can be broken in less than 24 hrs. though this algorithm is proven to be practically secure by the method of triple DES, still there are theoretical attacks. Hence there is the next version of DES i.e AES has evolved. The AES is nothing but the block cipher, transforms 128-bit data blocks under 128, 192, 256 bit secret key using both permutations and substitution like such mathematical operations, for further security, the National Institute of Standards and Technology (NIST) has called for the further development of algorithms regarding AES (Advanced Encryption Standard), after many such rounds that it has selected Rijndael as the next improvement that is used to encrypt the symmetric key and which is used in AES encryption of larger data.

2.1 Mathematical Procedure
In AES a large portion of the tasks are performed at byte level, with bytes speaking to components in the limited field GF (28). Remaining tasks are characterized as far as 4-byte words. First of the method includes the Galloy's Field Transformation GF(2^8)
The byte esteem in AES is spoken to as the accumulation of bits isolated by comma as {b7, b6, b5, b4, b3, b2, b1, b0}. These are spoken to as limited field components utilizing polynomial portrayal as:
\[ b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 \]
Every one of the tasks that are being utilized in AES calculation is clarified in the accompanying areas.

2.1.1 Polynomials with coefficients in GF(2^8). Polynomial in limited field components is given as:
\[ a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \]
Where coefficients of the above condition frames a word and are substantial bytes. The expansion of such polynomials is like that of our bit expansion. \[ b(x) = b_3x^3 + b_2x^2 + b_1x + b_0 \] be the second four term polynomial. Option is finished by including the coefficients of polynomial with like forces of x. This expansion is the XOR activity between the comparing BYTES in every polynomial.
Accordingly, \[ a(x) + b(x) = (a_3 \oplus b_3) x^3 + (a_2 \oplus b_2) x^2 + (a_1 \oplus b_1) x + (a_0 \oplus b_0) \]
Increase is determined utilizing:
\[ c(x) = c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \]
2.2 Algorithm

For the two its Cipher and Inverse Cipher, the AES calculation utilizes a round capacity that is made out of four distinctive byte-situated changes:

- Byte substitution
- Shift columns
- Mixing the information inside every segment of the State exhibit

**Add Round Key.**

![Flow chart of AES](image)

AES is a uses block cipher with block size 128 and key length is not fixed. Laying on the size of the key, rounds are calculated. And for each round all the above mentioned operations are performed. The result of each round is fed as an input to the next round[4]. **Note: In the final round, the Mix Column operation is omitted.**

### 3. Operationa and results

#### 3.1. Pre-Round

It is the first operation in the encryption mechanism. It is basically an 128 bit XOR operation. In this data input of 128 bits is XORed with user defined key of size 128 bits

Example:

- Input = 3925841d02de09fbdce118597196a0b32
- Key = 2b7e151628a809cf43c
- Output <= input XOR key;
3.2 S-Box
Substitution box i.e the substitution of rows and columns. It is used in sub-byte the next operation. It is a predefined table that has 128 hexadecimal values. These values are given input through the code. The substitution of row and column value would give the output value which is the intersection of that particular row and column. [3][5]
it can map all the input 128 bits and gives the table values for the given input i.e. every byte in the state is replaced by another.

3.4. **Shift Row** For the satisfaction of encryption perform moving of lines… A transportation step where each line of the state is moved consistently with various counter balances. A move push has 128 piece input and 128 piece yield. Each 128 piece is put away as a 4x4 network. Row1 is moved over c1 bytes, row2 over c2 bytes, and row3 over c3 bytes. The estimations of c1, c2, c3 rely upon the square length.

Sub Byte O/p:Shift Row i/p Shift Row Output

| a0.0 | a0.1 | a0.2 | a0.3 | a1.0 | a1.1 | a1.2 | a1.3 | a2.0 | a2.1 | a2.2 | a2.3 | a3.0 | a3.1 | a3.2 | a3.3 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| [a0.0] | [a0.1] | [a0.2] | [a0.3] | [a1.0] | [a1.1] | [a1.2] | [a1.3] | [a2.0] | [a2.1] | [a2.2] | [a2.3] | [a3.0] | [a3.1] | [a3.2] | [a3.3] |

A graphical representation of this operation can be found below:
3.5. Mix Column
This is performed on each column, by treating it as a polynomial (given below). The columns are considered as polynomials over GF($2^8$) and multiplied modulo $x^4 + 1$ with a fixed polynomial $a(x)$, given by $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. This can be described as…..

3.6. Key Generator
The key generator circuit functions to generate unique key for every round operation in AES algorithm. It is the main module in which the key is generated for the data that has to be encrypted it is thus done for all the different rounds involved. This operator is used to generate a unique key for every round operation in AES. It basically follows 5 steps:
Step 1: consider the last column of the user defined key.
Step 2: perform sub-byte operation for the step 1 output.
Step 3: perform left shift operation to 1 position.
Step 4: adding the RCON (round constant) values to step 3 output.

User defined is fed as an input to Key expander circuit to find the key generated output. To enhance the speed of the process, pipelining preferable to use for key generation. Number of keys generated is 160 byte for 10 rounds (excluding the pre rounds). In other words, number of rounds is based on key size. [12].

Figure 1: Simulation result
Figure 10 Simulation result

Figure 11: Simulation output of Mix column

Figure 12: Key generation result
3.7 Add Round

The primary function of Add round is to associate key generator output. Add round output is the XOR of key generation output and shift row or mix column output. After 1st round the add round result is again fed to the sub byte for 2nd round and the process repeats until it completes 10 rounds of operation. This is primarily done 10 rounds to complete all the 128 bits of data. The add round output for the first round is shown as:

![Add round key output](image1)

Figure 13. Add round output

3.8 Different Rows

![Steps involved in different rows](image2)

Figure 14. Steps involved in different rows

3.9 General Round

![Steps of general round](image3)

Figure 15. Steps of general round

3.10 Final Round

![All steps in final rounds](image4)

Figure 16. All steps in final rounds
4. Decryption

It is the process of taking text, encoded or encrypted data and converting back to the original information. This method i.e un-encrypting is attained using proper codes or keys. Companies generally encrypt data to maintain the information safe and secure. A decryption key is required to crack the data, otherwise special software's maybe required to make the data readable. In encryption original text (data) is converted into intangible cipher text. Whereas, decryption does the opposite to encryption, i.e., form cipher text using a key we can get back our original information.

The advantage in AES (decryption) is there is the same key for encrypting and decrypting the data. To write the code for decryption – it’s the inverse of the modules in encryption. There would be few changes in the modules like shift-row or mix column operations.

![Figure 17. Decryption Simulation](image)

5. Results

![Figure 18. Internal connections](image)

For implementing on an FPGA which has clock pins, j tack cable (joint test action group cable), LAN connections, serial communication external devices, USB cable port, switches-8. Pin nos are
T10,T9,B9,M8,N8,U8,V8,T5. The LED’s are in 0-7 positions(o/p pins). Output can be received in binary or digital format. Diligent board is the software that is used for passing the code to the board or FPGA. Then in UCF(user constrained file) obtained from new source. NET “ i/p or o/p name ” LOC = “pin no on the board”;. All 8 have to be given this way. Then select implement the design option. Generate programming file. The output we got in binary as- 00111001 = 39. This software is free and is available online also. The output can also be through the LCD screen.

6. Conclusion

The presence of an encryption mark on the screen is a must for any user to visit the webpage i.e for their data security. Now encryption has therefore become a business standard. We may get a doubt regarding the sustainability of the established AES algorithm but there are constant checks and reviews on this and just like in the discovery process of this algorithm there are still checks on this and making it perfect. Compared with the DES standards of 56 bit key size our AES is 1021 times 128 bit keys more than the DES ones. In fact it may be true that RSA is more secure than AES but the complex procedure cannot be successfully implemented in the present day fast paced applications, hence we have adopted the RSA standards for securing the keys and further the AES for the encrypting it. This procedure is proven to be safer and more secure for sending and receiving data. Moreover, creators in Soliman and Abozaid (2011), Gielata et al. (2008) and Qu et al. (2009) investigate pipelining, sub-pipelining, and circle unrolling methods to build the recurrence and throughput of AES execution on FPGA. Collapsed parallel engineering is utilized by Rahimunnisa et al. (2014) to get high throughput. The idea of collapsing is utilized to improve the zone usage while keeping up high throughput. So also we can utilize different thoughts of actualizing AES in those procedures and making our data more verified and our lives less difficult. Consequently there is a great deal of extension and prerequisite and thus we have to fabricate better thoughts of utilizing this.

References

[1] Muhammad H. Rais and Syed M. Qasim "Efficient Hardware Realization of Advanced Encryption Standard Algorithm using FPGA", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.9, September 2009
[2] Deshpande, A.M. Deshpande, M.S. Kayatanavar, D.N. "FPGA implementation of AES encryption and decryption", IEEE Transactions, Print ISBN: 978-1-4244-4789-3 , Jun 2009.
[3] Marko Mali , Franc Novak and Anton Biasizzo "Hardware Implementation Of AES Algorithm", Journal of Electrical Engineering, VOL. 56, NO. 9, 2005, 265-269 7
[4] RajenderManteena, "A VHDL Implementetion of the Advanced Encryption Standard-Rijndael Algorithm", College of Engineering University of South Florida, 2004.
[5] S. Morioka and A. Satoh "A 10-Gbps Full AES-CryptoDesign with a Twisted BDD S-Box Architecture", IEEE Transactions on VLSI Systems, Vol. 12, No. 7, July 2004, pp.686-691.
[6] C-P. Su, T-F.Lin, C-T.Huang; and C-W. Wu, "A High-Throughput Low-Cost AES Processor", Communications Magazine, IEEE, Vol. 41, Issue: 12, December 2003, pp. 86-91.
[7] S. Morioka and A. Satoh, "A 10 Gb/s full-AES crypto design with a twisted-BDD S-box architecture," in Proc. IEEE Int. Conf. Computer Design: VLSI in Computers and Processors, Freiburg, Germany, Sep. 2002, pp. 98-103.
[8] NIST, "ADVANCED ENCRYPTION STANDARD (AES,Rijndael)", FIPS-197, November 2001
[9] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, "Efficient implementation of Rijndael encryption with composite field arithmetic," in Proc. Cryptograph. Hardware Embedded Syst., Paris, France, May 2001, pp. 171-184.
[10] I. Verbauwhede, P. Schaumont, and H. Kuo, "Design and Performance Testing of a 2.29- Gb/s Rijndael Processor," IEEE J.Solid-State Circuits, Vol. 38, No. 3, Mar. 2003, pp. 569-572.
[11] N. Sklavos and O. Koufopavlou, "Architecture and VLSI implementation of the AES- proposal Rijndael," IEEE Trans. Comput., vol. 51, no. 12, pp. 1454-1459, Dec. 2002.
[12] V. Fischer and M. Drutarovsky, "Two Methods of Rijndael Implementation in reconfigurable Hardware", Proc. CHES, Vol. 2162, 2001, pp.81-96.
[13]  William Stallings "Cryptography and Network Security - Principles and Practices" fourth edition.