Data Encryption and Decryption Using Triple AES on FPGA

Asha. P,
PG Student
Dept. of ECE (VLSI & Embedded System Design)
Dr. Ambedkar Institute of Technology,
Bengaluru-56, India
ashateju.in@gmail.com

Dr.G.V.Jayaramaiah
Professor
Dept. of ECE
Dr. Ambedkar Institute of Technology
Bengaluru-56,India
gvjayaram@gmail.com

Abstract

Securing the digital data is of utmost important for digitisation. There are many encryption and decryption algorithms enabling such digital data security. However there still is a scope for improvising on the existing algorithm and to develop new algorithms. Triple Advance Encryption System (TAES) is one of the algorithms aiming for increased security of digital data. Intent of this paper is to elaborate on the key features, design and advantages of Triple Advance Encryption System (TAES) which varies from AES by using two different cipher keys for encryption.

Index Terms: *Triple Advanced Encryption Standard, look-up table, Rijndael.*

INTRODUCTION

Cryptography is the science of creating and using a cipher to prevent the unintended recipient(s) from using the secured information. Basic requirement of a strong encryption algorithm is that an individual who has knowledge of the algorithm and may be able to access more than one cipher texts shall not be able to decipher cipher text or to find the way to have the key. Various technologies and algorithms have been built to secure data and protect it from assorted hackers and fraudulent admittance. National Institute of Standards and Technology[1] (NIST), a branch of the US government, in the year 1997 decided to search for new algorithm as it was discovered that Data Encryption System(DES)[2] was not secure enough. NIST branch of US government in the year 2000officially adopted the AES algorithm later accepted it as a federal information processing standard under the FIPS-197[3].

The triple AES encryption is a symmetric block cipher using two different keys where sender and receiver both uses same two keys and undergoes multiple encryption. Plaintext of 128 bits and each key length of 128 bits are accepted as input. The algorithm takes ten rounds to mixes the data re-encrypting. Designing high speed architecture for AES [4]-[5] is very essential for designing the high-speed hardware implementation of the Triple Advanced Encryption Standard algorithm. It was the traditional method to implement the Sub Bytes and Inv Sub Bytes transformations by using[6]look-up tables approach, the proposed design deals with implementation of Sub byte and Inv Sub Byte by using combinational logic only. By this non LUT approach unbreakable delay incurred by look-up tables can be laminated. However this approach include inversion of Galois Field GF(2^8)[7]. It uses the concept of Rijndael’s AES Algorithm but has more number of keys to provide highly secured data transmission. Single symmetric key are used in AES algorithm which makes the data to be easily hacked by unauthorized person. So the proposed system works on using two different keys to encrypt the data so that all these keys are required to successfully decrypt the data. The Proposed system ensures that data cannot be cracked even though intermediate data appears. Section II discusses about proposed algorithm, Section III discusses about the implementation of TAES, Section IV represents Simulation Results, Section V describes Conclusion .

TRIPLE AES

Triple AES is a process of reuse of AES with serial installation of three instances of AES to improve the security of the data. In proposed system Encryption process deals with two different keys leads to formation of TAES. The AES operation is performed three times with two different keys. The two different keys such as K_1 and K_2 used on a plain text P to convert it into cipher text C.

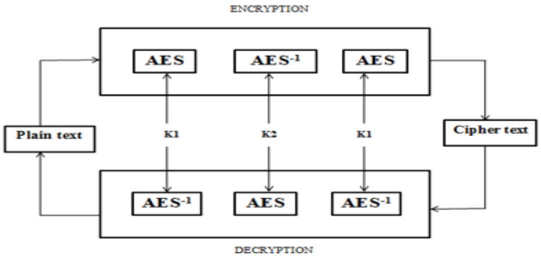


Fig 1. Triple AES-Block Diagram

By using the Rijndael algorithm, the first stage of encryption is done with the help of key K_2 and the result of this part is fed to next block of encryption having key $K_2$and third encryption is done with the help of key K_1. Multiple encryptions of the data increase its security and make it difficult to crack the data. Fig 1 shows the model for implementing TAES. The decrypting block is similar as encryption procedure but operates in inverse order.

TAES ALGORITHM IMPLEMENTAION

The TAES algorithm is a block cipher, in which both sender and receiver use two keys

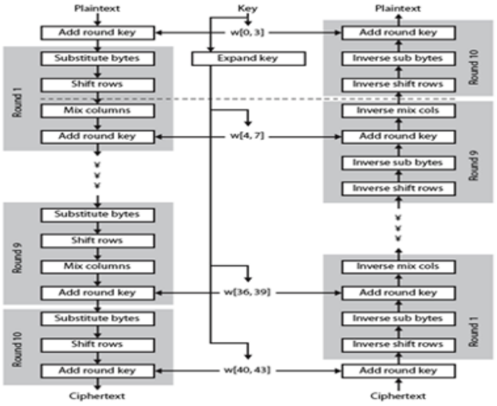


Fig.2. Encryption and decryption process of Triple AES algorithm

Encryption and decryption and data block consist of a128 bits which is accepted as 4*4 array of bytes which is named as state, on which the basic operations of the TAES algorithm such as Sub byte, Mixed Column, Shift Rows and Add Round key are performed[9]. Fig 2 shows the encryption and decryption process of TAES.

SubByte

This section illustrates implementation which is the composition of multiplicative inverse followed by affine transformation for InvSubByte transformation,before computingmultiplicative inverse, the inverse affine transformation is done. The same multiplicative inverse can be used for Inv. Multiplicative inverse can be calculated by below equation

$$(bx + c) = b(bB + bcA + c)x + (c + bA)(bB + bcA + c)$$

The above equation include operations such as multiplication ,addition, squaring and inversion in GaliosField GF(2^8).By using simplified equation multiplicative inverse module can be constructed as shown in figure 3

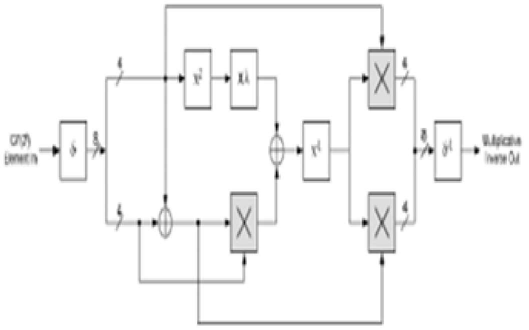


Fig.3.*Multiplicative Inversion model of Sbox*

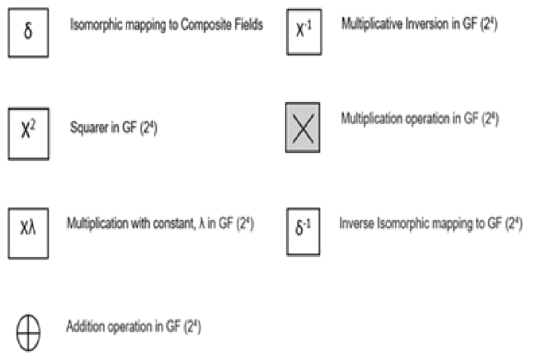

δ
Isomorphic mapping to Composite Fields
x⁻¹
Multiplicative Inversion in GF (2⁴)
x²
Squarer in GF (2⁴)
Multiplication operation in GF (2⁴)
xλ
Multiplication with constant, λ in GF (2⁴)
δ⁻¹
Inverse Isomorphic mapping to GF (2⁸)
Addition operation in GF (2⁴)

Fig 4.*Building blocks within the multiplicative inversion module*

Shift Row

In Shift row transformation operates on individual row of nation .It cyclically moves the bytes in each Row of state toward left by way of precise offset price. the primary row is unaltered, the second row is moved by1byte, the third is altered through 2 byte and similarly fourth row is shifted with the aid of 3 byte position.

Mix Columns

In Mix Columns transformation [8] performs column wise operation by using an invertible linear transformation technique. Mix Column transformation in Cipher that takes each and every columns of State and mix their data individually of one another to produce new columns using GF (2^8) polynomial.

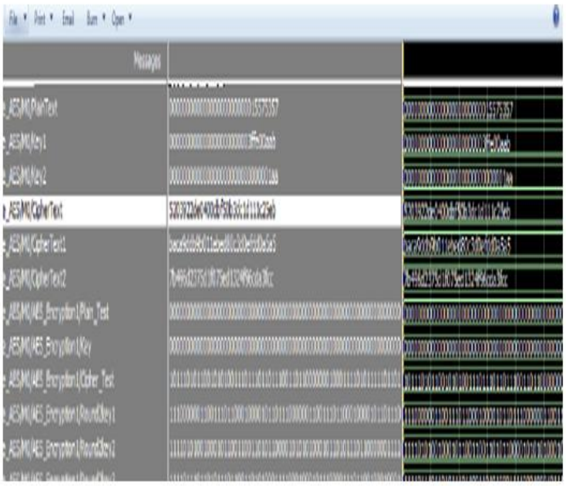
Add Round key

In Add round key transformation around key which is taken from key scheduling process is added to output of mixed column by simple bitwise operation and output is 4byte of column at a single time This process is evaluated as a column wise addition operation between the single column of state and word of the round key, it can also be seen as a byte-level operation.

SIMULATED RESULTS

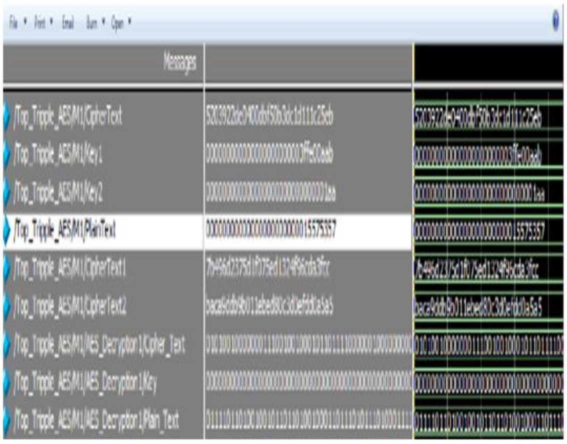
We synthesize the Virology components by means of XST tool Xilinx ISE Project Navigator Xilinx ISE 14.7 besides the language used, when the first phase finishes with the first block to be encrypted, this next block goes into that phase while the first block goes into the next operation. Finally, we have used parallelism of code for calculating total cipher text or original plain text. Xilinx-14.7 is used for synthesizing the proposed system and the simulation is done in Xilinx ISE Simulator. The simulation result for 128 bit key using triple AES is computed by Xilinx ISE 14.7

All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption with less no of slices



```
_AES(M)/PlainText                     [illegible]
_AES(M)/Key1                          [illegible]
_AES(M)/Key2                          [illegible]
_AES(M)/CipherText                    [illegible]
_AES(M)/CipherText1                   [illegible]
_AES(M)/CipherText2                   [illegible]
_AES(M)/AES_Encryption1/Plain_Text    [illegible]
_AES(M)/AES_Encryption1/Key           [illegible]
_AES(M)/AES_Encryption1/Cipher_Text   [illegible]
_AES(M)/AES_Encryption1/RoundKey1     [illegible]
_AES(M)/AES_Encryption1/RoundKey1     [illegible]
```

Fig 5. TAES Encryption output



```
/Top_Tripple_AES/M1/CipherText                    5203922de0400dbf50b3dc1d111c25eb
/Top_Tripple_AES/M1/Key1                          000000000000000000000003ffe00aeb
/Top_Tripple_AES/M1/Key2                          00000000000000000000000000001aa
/Top_Tripple_AES/M1/PlainText                     0000000000000000000000015575357
/Top_Tripple_AES/M1/CipherText1                   7b496d2375d1f075ed1324f96cda3fcc
/Top_Tripple_AES/M1/CipherText2                   baca9ddb9b011ebed80c3d0efdd0a5a5
/Top_Tripple_AES/M1/AES_Decryption1/Cipher_Text   0101001000000011100100100010110111100000010000000
/Top_Tripple_AES/M1/AES_Decryption1/Key           0000000000000000000000000000000000000000000000000
/Top_Tripple_AES/M1/AES_Decryption1/Plain_Text    0111101101001001011011010010001101110101110100011
```

Fig 6. TAES Decryption output

CONCLUSIONS

Due to the needs in communications for safer and more secure, a new variation of AES has been proposed and implemented. In this paper, the TAES using two different keys has been designed and results are verified. The performance comparison of TAES with different encryption standards are obtained in term of time to crack, security, and resources utilization. The throughput is increased because of the greater input block and the security of the algorithm is achieved due to larger key size. The area requirement is more but it can be neglected. TAES is best for applications in which very high security and increased throughput is required such as in multimedia communications.

SCOPE AND FUTURE DEVELOPMENT

For the foreseeable future TAES is an excellent and reliable choice for the security needs of highly sensitive information. The AES will be at least as strong as TAES and probably much faster. It's the industry mandate from Visa and MasterCard that's requiring ATM deplorers to upgrade and/or replace their legacy terminals. In a nutshell, it's all about three waves of encryption, and it's designed to make ATM transactions more secure.

REFERENCES

1. *Advanced Encryption Standard (AES)*, Nov. 26, 2001.
2. A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar. An FPGA implementation /and performance evaluation of the AES block cipher candidate algorithm finalist. presented at *Proc. 3rd AES Conf. (AES3)*
3. V. Fischer and M. Drutarovsky, “Two methods of Rijndael implementation in reconfigurable hardware,” in *Proc. CHES 2001*, Paris, France, May 2001, pp. 77–92.
4. K. Gaj and P. Chodowiec. Comparison of the hardware performance ofthe AES candidates using reconfigurable hardware. presented at *Proc.3rd AES Conf.(AES3).* [Online].Available:http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html
5. H.Kuo and I.Verbauwhede, “Architectural optimization for a 1.82 Gbits/sec VLSI implementation of the AES Rijndael algorithm,” in *Proc. CHES 2001*, Paris, France, May 2001, pp.51–64.
6. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact Rijndael hardware architecture with S-Box optimization,” in *Proc. ASIACRYPT2001*, Gold Coast, Australia, Dec. 2000, pp. 239–254.
7. 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. CHES 2001*, Paris, France, May 2001, pp. 171–184.
8. G. P. Saggese, A. Mazzeo, N.Mazocca, and A.G.M. Strollo,“An FPGA based performance analysis of the unrolling, tiling and pipelining of the AES algorithm,” in *Proc. FPL 2003*, Portugal, Sept. 2003.

9. X. Zhang and K. K.Parhi,“Implementation approaches for the advanced encryption standard algorithm,” *IEEE Circuits Syst. Mag.*, vol. 2, no. 4,pp. 24–46, 2002.