International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

High Throughput AES Algorithm with s-box sharing for Threshold Implementation

P.V.V Deepika[1]& P.Soundarya Mala[2]
[1]PG student (VLSI & Embedded Systems), GIET, Rajahmundry, India [2]Associate Professor, ECE Dept, GIET, Rajahmundry, India

[1]putchaladeepika@gmail.com; [2]palivela.soundarya@gmail.com

ABSTRACT—

A high speed security algorithm is always important for wired/wireless environment. Fast evaluation of digital data exchange occurs in recent years. Day by day need for security services and security mechanism are increasing. The use of cryptography requires the handling of secret keys. Therefore, new approaches are necessary to fulfill the requirements in both software and hardware implementation approach i.e., advanced encryption standard (AES). AES is one of the best security algorithm to provide data security by using special techniques i.e., "S-box sharing & threshold implementation", AES can also provide input data length upto 256 bits which is four times greater than the DES. This leads to the increase in throughput i.e., the obtained by AES 256 key length is >60 kbps where as the throughput obtained by AES 128 key length is 47kbps only. Therefore AES with 256 key length is well suitable for security purposes. The logic verification & synthesization is done through Xilinx tool.

Index Terms– cryptography; DES; AES; S-box; Threshold Implementation.

I. INTRODUCTION

Recent developments in information technologies made the secure transmission of digital data a critical design point. Large data flows have to be exchanged securely and involve encryption rates that sometimes may require hardware implementations. For this implementations we are using (FPGA's) Field programmable gate arrays . Nowadays cryptography has a main role in Embedded systems designs. In many applications, the data requires a secured connection which is usually achieved by cryptography.

Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality ofcommunication through an insecure channel. It protects against unauthorizedparties by preventing unauthorized alteration of use. It uses an cryptographicsystem to transform a plaintext into a cipher text, using most of the time a key.There are two basic types of encryption commonly used today, symmetric key(sender and receiver shares the same key) and asymmetric key (sender and receiver shares different key) encryption. All of the cryptographic algorithms we have looked at so far have some problem. The earlier ciphers can be broken with case on modern computation systems. Symmetric systems contains Data Encryption Standard (DES), 3DES, and Advanced Encryption Standard (AES) use an identical key for the sender and receiver. DES (Data Encryption Standard) was considered as the standard of symmetric key encryption, which has a key length of 56 bit. This key length is considered as very small and can be easily broken. The National Institute of Standards and Technology(NIST), proposed Rijndael algorithm as the Advanced Encryption Standard(AES) in October 2000 providing strong security and high flexibility. AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits.

A. Review of threshold implementation

It is a provably secure countermeasure, Against (first) order power analysis based on *multi party computation* and *secret sharing. the countermeasures are nothing but*

*a)*Hardware countermeasures: Balancing power consumption.

*b)*Masking: Randomizing intermediate values, There should Implementations, Shamir's Secret Sharing.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

c)Leakage-Resilient Crypto

The Threshold Implementation (TI) countermeasure was proposed by Nikova .TI applies secret-sharing to achieve provable resistance against first order side channel attacks if the following three requirements are fulfilled.

1) Correctness: Correctness means that combining the output of the different shares retrieves the original output in a correct way.

2) Non-completeness: Non-completeness means that the equation used to evaluate any output share should be missing at least one input share. This requirement enforces that the information required to compute the secret value (all the shares) is not present in the system at any time instant. Hence, any vulnerability in the implementation (e.g. glitches) cannot leak the secret key.

3) Uniformity: If the input shares are uniformly distributed, the output shares must also be uniformly distributed.

B. Review of s-box sharing

In cryptography, an s-box(substitution-box) is a basic component of symmetric key algorithms which performs substitution. In block ciphers, they are typically used to obscure the relationship between key and the cipher text. In general, an s-box takes some number of input bits, m, and transforms them into some number of output bits, n, where n is not necessarily equal to m. An m*n s-box can be implemented as a lookup table with 2*m words of n bits each .Fixed tables are normally used, as in the data encryption standard, but some ciphers the tables are generated dynamically from the key. We have so many s-box implementations. Those are raw implementation, adjusted implementation and nimble implementation. Here we are implemented with one of the s-box implementation that is nimble implementation. These implementations are implemented with multipliers, inverters and registers. From this we conserve the uniformity property and the security of each block is achieved only by the correctness and non completeness properties, we can discard the uniformity property and implement these nonlinear functions with the smallest number of shares. This construction requires only 32-bits of extra randomness per s-box calculation, including the number of shares for the s-box input.

C. Features

The features are mainly area and randomness. From these we got throughput. Threshold implementation also allows features between area, randomness, and the resistance against differential power analysis. In the nimble implementation reduces the area cost significantly compared to the uniformly shared raw and adjusted implementations.

S-box	*Raw*	*Adjusted*	*Nimble*
Multiplier	*625/458*	*625/458*	*418/308*
Inverter	*618/490*	*618/490*	*594/375*
Registers	*725*	*661*	*576*

Table I: Synthesis Results For Different Versions of S-Box TI with Compile Commands

We show the area, randomnessrequirements, and timings of AES implementations .We again provide results using the same compilation techniques as the S-box implementations and compare them with the old results. The area costs for the state and the key arrays include the ANDs and XORs. In our implementations, the S-box occupies (30-40) % of the total area. Compared to the implementation S-boxes with uniform blocks are 13% smaller and S-box with non uniform blocks is 33% smaller. These results show a significant area and randomness improvement of the nimble implementation.

Design	*Old results*	*Raw*	*Adjusted*	*Nimble*
State array	*2529*	*1698*	*2473*	*1687*
Key Array	*2526*	*1890*	*2762*	*1844*
S-box	*4244*	*3708/3004*	*3653/2949*	*2835/2224*
Mix columns	*1120*	*770/544*	*1156/816*	*770/544*
Control	*255*	*242*	*242*	*242*
Key XOR	*64*	*48*	*72*	*48*
MUXes	*376*	*746*	*853*	*693*
Total	*11114/11031*	*9102/8172*	*11221/10167*	*8119/7282*
Cycles	*266*	*246*	*246*	*246*
Randomness	*48*	*44*	*44*	*32*

Table- II: Synthesis Results for Different Versions Of AES TI With Compile Commands

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

II. ALGORITHM OF ADVANCED ENCRYPTION STANDARD – 128

A. AES algorithm

AES was released by NIST in 2001 as the new-generation data encryption standard for use in the USA, and it was adopted as an ISO international standard in 2005. Nowadays, smartcards are widely used in many real-life security applications as a medium of authenticating the identity of a user and/or executing cryptographic operations. Being used more and more widely in reality. One of the most widely used software countermeasures is the Boolean masking method. The main idea of thismethod is to mask any sensitive intermediate value by XORing it with one ormore randomly generated values called masks. Another kind of software countermeasures is hiding, such as randomizing the operations of an algorithm.

B. Basic Block Diagram Of AES

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different in a number of ways. A number of AES parameters depend on the key length. Like 128, 160, 192, 224, 256**,** *butpreferable key lengths are 128,192,256.* For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. The algorithm have transformations of four stages. They are Substitute bytes, Shift rows, Mix Columns, Add Round Key for encryption. For decryption these transformations are in reverse, Inverse Shift rows, Inverse Substitute bytes, Inverse Add Round Key, Inverse Mix Column. These transformations are applied for both encryption and decryption in each stage of round.

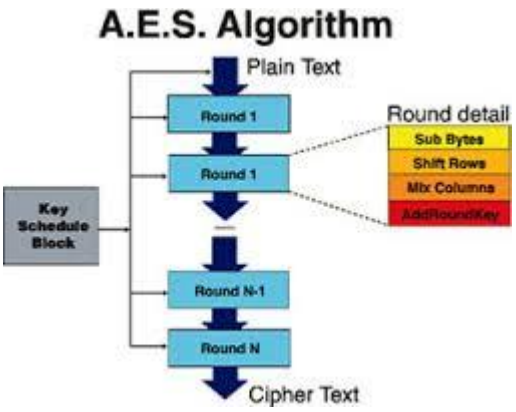

Fig 1: Block diagram of AES

The algorithm begins with an **Add round key** stage followed by 9 rounds of four stagesand a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of it's counterpart in the encryption algorithm. The tenth round leaves out the **Mix Columns** stage. The first nine rounds of the decryption algorithm consist of inverse of four transformations. Again, the tenth round simply leaves out the **Inverse Mix Columns** stage. From this AES-128 we got throughput 47kbps only.

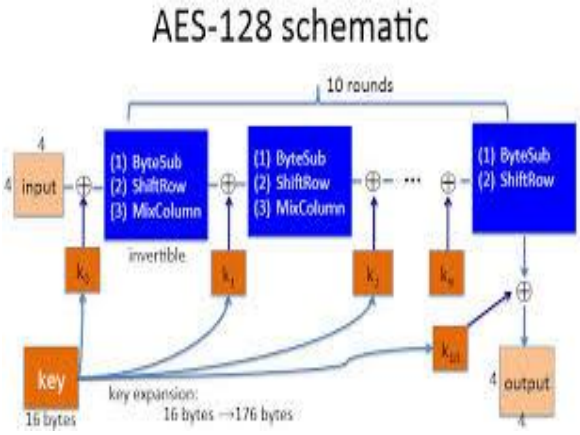

Fig2: AES schematic

III. ALGORITHM OF ADVANCED ENCRYPTION STANDARD-256

A.AES algorithm

Here also we have same type of transformation of operations and rounds of operations are changed. For the improving block ciphers we are taken this AES-256 algorithm. Block ciphers are nothing but increasing key length, variable number of rounds and so on**.***In AES-256 algorithm we have 14 rounds of with the type of transformations in each stage. Those transformation of operations are done for both encryption and decryption process also. Those are nothing but substitute bytes***,** Shift rows, Mix Columns, Add Round Key and Inverse Shift rows, Inverse Substitute bytes, Inverse Add Round Key, Inverse Mix Columns.

B.Block diagram of AES-256

The AES-256 algorithm begins with an **Add round key** stage followed by 14 rounds of four stagesand a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of it's counterpart in the encryption algorithm. The fourteenth round leaves out the **Mix Columns** stage. The first thirteen rounds of the decryption algorithm consist of inverse of four transformations. Again, the tenth round simply leaves out the **Inverse Mix Columns** stage. From this AES-256 we got throughput more than 60kbps. Throughput is a

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

measure of how many units of information of a system process in a given amount of time. It is applied broadly to systems ranging from various aspects of computer and network system to organizations.

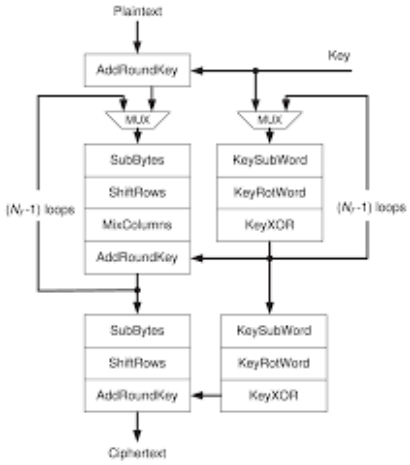

Fig 3: *Process of AES algorithm*

(a)Substitute Bytes:

This stage (known as SubBytes) is simply a table lookup using a 16×16 matrix of byte values called an **s-box**. This matrix consists of all the possible combinations of an 8 bit sequence (28 = 16 × 16 = 256). However, the s-box is not just a random permutation of these values and there is a well defined method for creating the s-box tables. The s-boxes are made up and can simply take them as table lookups. Again the matrix that gets operated upon throughout the encryption is known as **state.** We will be concerned with how this matrix is effected in each round.

Table-III
S-box table

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Table- IV
Inverse s-box table

X \ Y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	9	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	8	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	0	8c	bc	d3	0a	f7	e4	58	5	b8	b3	45	6
7	d0	2c	1e	8f	ca	3f	0f	2	c1	af	bd	3	1	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1a
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	7	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	4	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

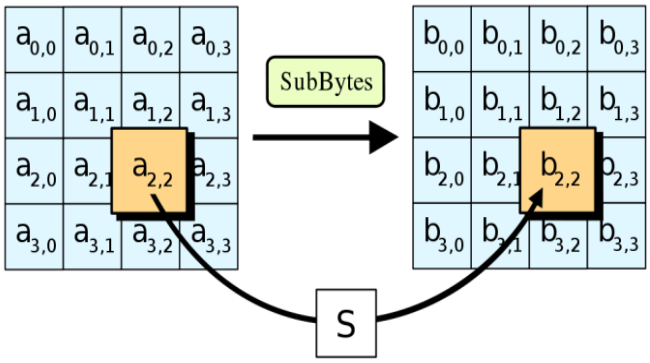

Fig 4*: Substitute Bytes Stage of the AES algorithm.*

(b)Shift Row Transformation:

This is a simple permutation. The first row of **state** is *not* altered. The second row is shifted 1 bytes to the left in a circular manner. The third row is shifted 2 bytes to the left in a circular manner. The fourth row is shifted 3 bytes to the left in a circular manner. The Inverse Shift Rows transformation (known as InvShiftRows) performs these circular shifts in the opposite direction for each of the last three rows.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

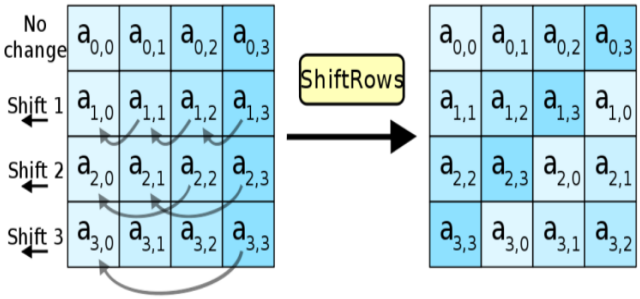

Fig 5:*Shift Rows stage*

(c)Mix Column Transformation:

This stage (known as Mix Column) is basically a substitution. Each column is operated on individually. Each byte of a column is mapped into a new value that is a function of all four bytes in the column.The transformation can be determined by the following matrix multiplication on **state.** For inversemix column transformation also done like this but in reverse case.

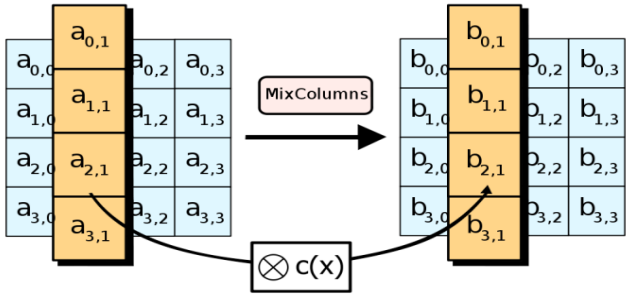

Fig 6*: Mix Columns operates on the State column-by-column*

The mix column transformation can be done by using XOR operation. After that we have to multiply with some parametrized matrix. This matrix is irreducible polynomial function. This is the standardized form of this transformation.

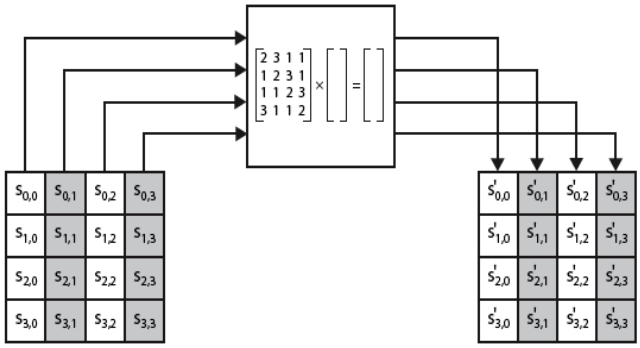

Fig 7*: Mix Columns stage*

(d)Add Round Key Transformation:

In this stage (known as Add Round Key) the 128 bits of **state** are bitwise XORed withthe 128 bits of the round key. The operation is viewed as a column wise operation between the 4 bytes of a **state** column and one word of the round key. This transformation is as simple as possible which helps in efficiency but it also effects every bit of **state**.

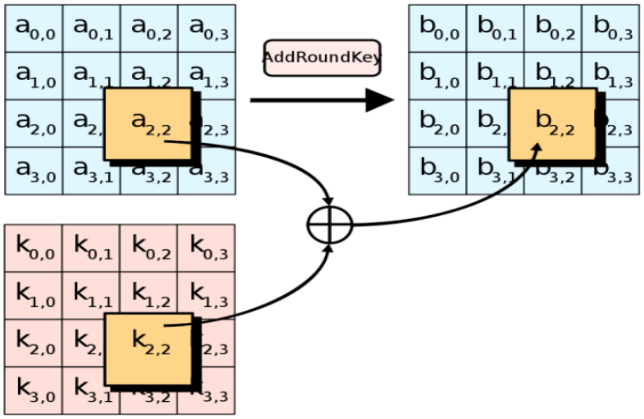

Fig8: *Add Round Key XORs each column of the State with a word from the key schedule.*

IV. RESULTS AND DISCUSSION

A. Simulation Environment:

By making use of the waveforms, it is very easier to evaluate the effectiveness of the proposed algorithm through simulation. Modelsim is used for simulation and Xilinx is used for evaluating the time, area and throughput for the existing And

International Journal of Research
International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

proposed architecture.

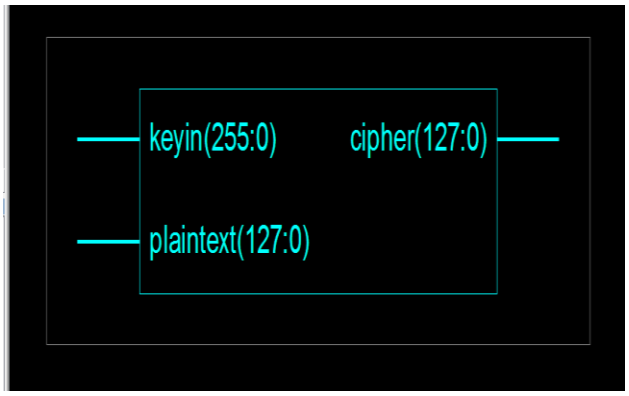

Fig 9: *Block diagram of encryption*

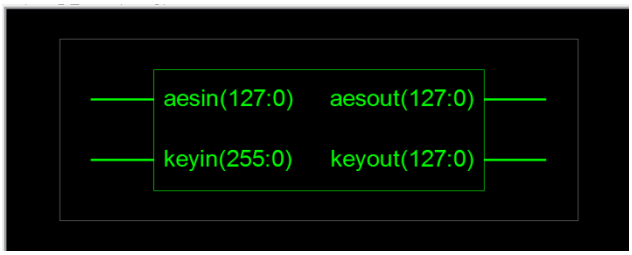

Fig 10: Block diagram of decryption

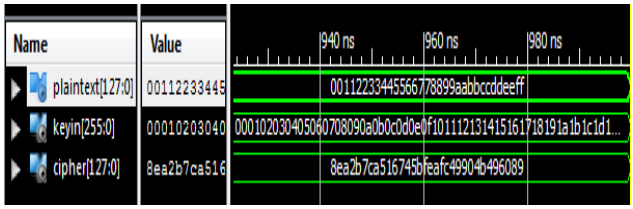

Fig 11: Simulation result of Encryption

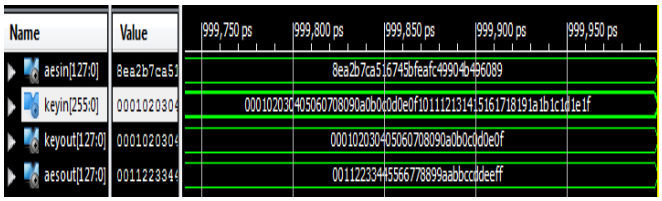

Fig 12: Simulation result of Decryption

B. Performance Analysis:

Table-v
Result analysis

METHOD	*THROUGHPUT FORMULAE*	*THROUGHPUT*
AES-128	*Delay*input[input: 128]*	*47KBPS*
AES-256	*Delay*input[input : 256]*	*60KBPS[more than 60KBPS]*

V. CONCLUSION

This paper shows successful implementation of AES 256 bit algorithm. It provides better security from unauthorized access. The data was encrypted using different keys and the original data was retrieved via decryption of cipher text. This algorithm guarantees secure end to end transfer of data without any corrupt data. It gives high throughput rate i.e., >60kbps than the throughput obtained by AES 128 bit is 47kbps. As a result the AES algorithm well performed in terms of speed and performance of security. The timing analysis verified on Xilinx simulator. The xilinx13.2i software is utilized in the project and code is written on VHDL. In future the implementation may beimplemented in military and forensic laboratories.

ACKNOWLEDGEMENT

I would like to thank my guide and all the reviewers for their constructive comments and active support to my work.

REFERENCES

[1] BegülBilgin, BenediktGierlichs, SvetlaNikova, VentzislavNikov, and Vincent Rijmen,vol. 34, NO. 7, JULY 2015

[2] P. C. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology—CRYPTO* (LNCS 1666). Berlin,GermanySpringer, 1999, pp. 388–397

[3] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*, AES Algorithm Submission, September 3, 1999.

[4] L. Goubin and J. Patarin, “DES and differential power analysis the ‘duplication’ method,” in *Cryptographic Hardware and EmbeddedSystems* (LNCS 1717). Berlin, Germany: Springer, 1999, pp. 158–172.

[5] J. Daemen and V. Rijmen, *The block cipher Rijndael*, Smart Card research and Applications, LNCS 1820, Springer-Verlag, pp. 288-296.

[6] T. S. Messerges, "Securing the AES finalists against power analysis attacks," in *Fast Software Encryption* (LNCS 1978). Berlin, Germany: Springer, 2000, pp. 150–164.

[7] J. Daemen and V. Rijmen, *The block cipher Rijndael*, Smart Card research and Applications, LNCS 1820, Springer-Verlag, pp. 288-296.

[8] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare, "FPGA Implementation of AES Algorithm", International Conference on Electronics Computer Technology (ICECT), pp. 401-405, 2011 3rd.

[9] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[10] Ahmad, N.; Hasan, R.; Jubadi, W.M; "Design of AES S-Box using combinational logic optimization", IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 696-699, 2010.