International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 3 Issue 5‖ May 2014 ‖ PP.01-07

FPGA Based Design of AES with Masked S-Box for Enhanced Security

[1,]Lekshmi R , [2,] Sajan Xavier,
[1,]*M.Tech Student, Nehru College Of Engineering and research centre, Pampadi*
[2,]*Assistant Professor Nehru College Of Engineering and research centre, Pampadi*

ABSTRACT*: Increasing need of data protection in computer networks led to the development of several cryptographic algorithms hence sending data securely over a transmission link is critically important in many applications. In order to protect "data-at-rest" in storage area networks from the risk of differential power analysis attacks without degrading performance, a masked S-Box is proposed. However, this architecture requires large field programmable gate array (FPGA) resources. For optimizing the area for an AES ,we map its operations from GF(2^8) to GF(2^4) as much as possible. The LUT based design of S-box consumes almost 75% of power.The values of s-box are known to everyone. Masking each values in the s-box by another masking function increase the system security and reduces the side channel attacks. The masking module can be implemented on any part of AES algorithm and re-masking module is used to remove the mask. Masking module can be used to increase the system security.*

KEYWORDS*: Advanced encryption standard (AES), differential power analysis (DPA), field programmable gate array (FPGA), masking, Encryption; Decryption, Galois field.*

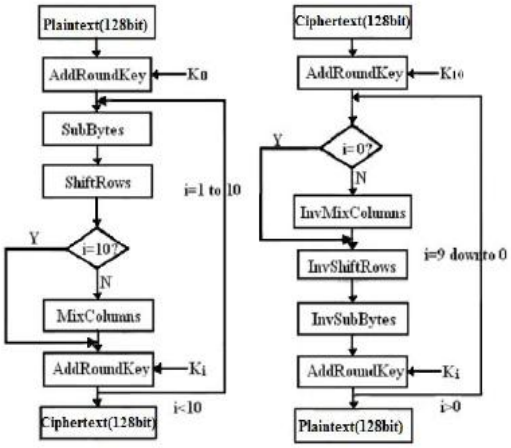
I. INTRODUCTION

WITH the development of information technology, protection of sensitive information via encryption is becoming more and more important to daily life and in 2001, the National Institute of Standards and Technology (NIST) selected the Rijndael algorithm as the Advanced Encryption Standard (AES) [1], which replaced the Data Encryption Standard (DES) [2]. Since then, AES has been widely used in a variety of applications, like secure communication system, high-performance database servers, digital video/audio recorders, RFID. The algorithm described by AES is a symmetric-key algorithm type, which means the same key is used for both encrypting and decrypting the data. the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits .Hardware implementation of cryptographic algorithms are physically secure than software implementations since outside attackers cannot modify these. In order to achieve higher performance in today's heavily loaded communication networks and others, hardware implementation is a wise choice in terms of better speed and reliability. Implementations of the Advanced Encryption Standard (AES), including hardware applications with limited resources (e.g., smart cards), may be vulnerable to\side-channel attacks" such as differential power analysis.One counter measure against side channel attacks is adding a random mask to the data; this randomizes the statistics of the calculation at the cost of computing \mask corrections. The Boolean masking is a good candidate to be applied to the AES in SANs, but if we directly apply it to the AES, one masked AES's S-box over GF(2^8) with two 8-bit input and output masks needs to store 2^8 × 2^8 × 256 bytes (16.8 Mbytes). Therefore, for a whole 128-bit masked AES with an unrolled architecture, it needs to store around 2952.8Mbytes. This is too big to be fit into any field programmable gate array (FPGA). To have a feasible FPGA implementation, one possible way is to transform the S-box computation from GF(2^8) to GF(2^4).Here, the related operations like the masked Mix Column, masked Add Round Key, and also masked Shift Rows including redundant masking values are all calculated over GF(2^4) .so we need to transform the input values from GF(2^8) to GF(2^4) and transform the output values back from GF(2^4) to GF(2^8) once which reduces hardware resources. The new masking scheme combines the concepts of multiplicative and additive masking in such a way that security against first-order side-channel attacks are maintained. As a result small implementations in dedicated hardware can be achieved.This paper presents two software implementations of the AES algorithm, and shows that AES implementations using masked S-Box of the same implementations can achieve high performance and security.

II. ADVANCED ENCRYPTION STANDARD

AES is a symmetric encryption algorithm, which takes a 128-bit data block as input and performs several rounds of transformations to generate output cipher text, where each 128-bit data block is processed in a

4-by-4 array of bytes which is called the state and the round key size can be 128, 192 or 256 bits and the number of rounds repeated in the AES called Nr is defined by the length of the round key, that is 10, 12 or 14 for key lengths of 128, 192 or 256 bits, respectively .Here fig. 1 shows the AES encryption steps with the key expansion process. In encryption process there are four basic transformations applied as follows:



. Fig.1.Block diagram of AES encryption and decryption.

1. SubBytes: The SubBytes operation is a nonlinear byte substitution operation. Each byte from the input state is replaced by another byte according to the substitution box (called the S-box).S-box is computed based on a multiplicative inverse in the finite field GF(2^8) and a bitwise affine transformation.
2.

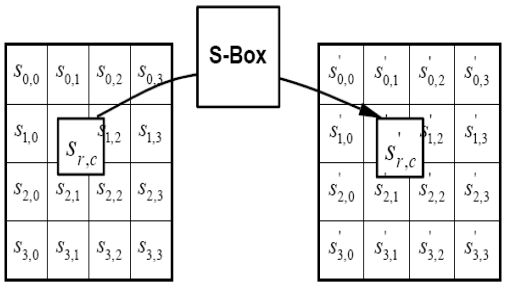


Fig 2.Sub bytes

2. Shift Rows: In the Shift Rows transformations, the first row of the state array remains unchanged whereas the bytes in the second, row is cyclically shifted by one bytes to the left, third row is cyclically shifted by two bytes to the left, and forth row by three bytes to the left..

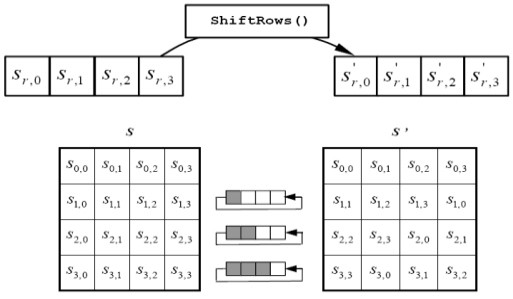


Fig 3.Shiftrows

3. Mix Columns: During the Mix Columns processes, each column of the state array is considered as a Polynomial over GF(2^8) field. After multiplying modulo x^4+1 with a fixed polynomial a(x), which given by a(x)={03}x^4 +{01}x^2+{01}x+{02},we gets the result as the corresponding column of the output state.

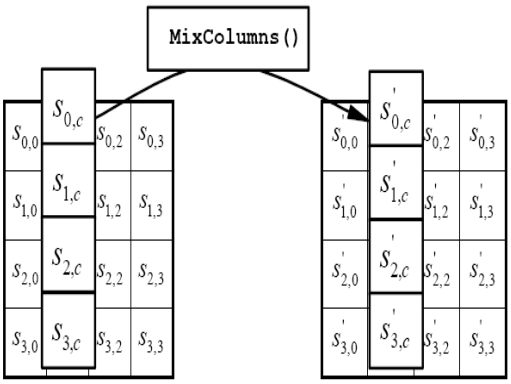


Fig .4.Mixcolumns

4. Add Round Key: A round key is added to the state array using a bitwise exclusive-or operations. Round keys are calculated in the key expansion process also If Round keys are calculated on the fly for each data block, then it is called AES with online key expansion.

For most applications, the encryption keys do not change as frequently as data. And ss a result, round keys can be calculated before the encryption process. Then it is kept constant for a period of time in local memory or registers which is called AES with offline key expansion.

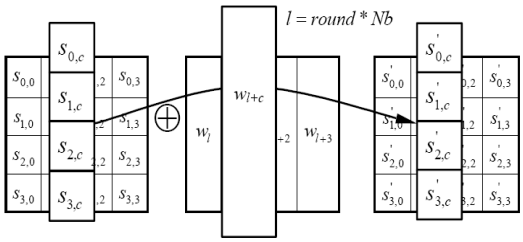


Fig.5.AddRoundkey

The three steps in each key expansion round are given below

[1] Key Sub Word: This operation takes a four byte input word and produce an output word by substituting each byte in the input to another byte according to the S-box

a. .

[2] .Key Rot Word: The function Key Rot Word takes a word $[a_3, a_2, a_{1,} a_0]$performs a cyclic permutation, and returns the word $[a_{0,} a_{1,} a_{2,} a_3]$as output.

[3] Key XOR: Here, every word w[i] is equal to the XOR of the previous word w[i]-1 and the word Nk positions earlier which is given by W[i-Nk],where Nk equals 4, for the key lengths of 128,6 for the key lengths of 192 and 8 for the key lengths of 256 bits.

For the decryption algorithm, it applies the inverse transformations in the same manner as the encipherment.

III. AES IMPLEMENTATIONS

In this section two different AES cipher implementations are described.

ONE-TASK ONE-PROCESSOR (OTOP)

This is the most straightforward implementation of an AES cipher which is to apply each step in the algorithm as a task in the dataflow diagram as shown in Fig. 6a. Each task in the dataflow diagram can be mapped on one processor on the targeted many-core platform. This implementation is called as one-task one-processor. And for simplicity, all of the execution delay, input rates, and output rates in the following dataflow diagrams are omitted .

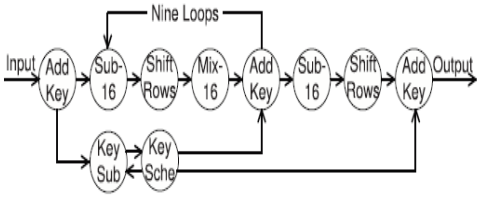


Fig.6.One-Task-One Processor Dataflow diagram

LOOP-UNROLLED NINE TIMES

In order to enhance the AES cipher's throughput, apply loop unrolling to the OTOP model and obtain the Loop-unrolled Nine Times dataflow diagram as shown in Fig. 7a. The loop unrolling breaks the dependency among different loops and allows the nine loops in the AES algorithm to operate on multiple data blocks simultaneously. In order to improve the throughput as much as possible, unroll the loops in both the AES algorithm and the key expansion process by Nr - 1 and Nr times, that equals 9 and 10, respectively.
After loop unrolling, throughput of the AES implementation is increased to 266 cycles per data block equals 16.625 cycles per byte.

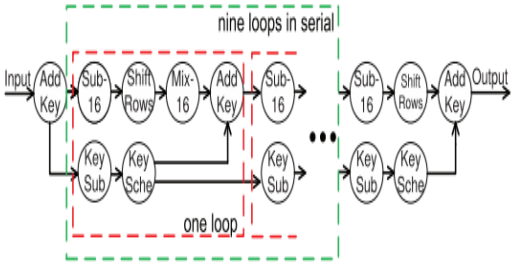


Fig.7.Loop-Unrolled Nine Time Dataflow diagram

IV. EXISTING SYSTEM

The security of sensitive information transmitted via the Internet has been the focus of modem cryptographist's attention. The Rijndael algorithm was adopted as the Advanced Encryption Standard (AES) by the American National Institute of Standards and Technology (NIST). Data Encryption Standard (DES) is the first open encryption algorithm by USA government to protect the sensitive information. However, the shorter length of key and the complementary property along with the existence of weak and semi-weak keys reduce the security of DES ,this is to find a stronger encryption algorithm to substitute the DES. The objective in using the AES is to transfer the data so that only the desired receiver with a specific key would be able to retrieve the original data.. The existing scheme of the S-box is linear and it is not secure against cryptanalysis. The existing S-Box is shown below.

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

Fig.8.Predefined S-Box

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	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	08	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	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	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	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	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	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Fig.9.predefined inverse S-Box

V. PROPOSED SYSTEM

Through the research of Rijndael algorithm, a non-linear layer of S-box transformation is a key to make the entire algorithm strong. The cryptographic strength of the AES depends on the choice of the S-box. Most of the cryptographists have discovered that there is some weakness in the design of the existing S-box. In order to improve complexity of S-box structure our approach is combining a dynamic nonlinear transformation method and linear function and a good S-box can be very well resist differential cryptanalysis, linear cryptanalysis attacks and so on. The Advanced Encryption Standard (AES) has S-boxes in it substitution Bytes and Inverse Substitution bytes. In order to enhance the complexity of the S-Box's structure, a masked S-Box is considered in the design of AES.

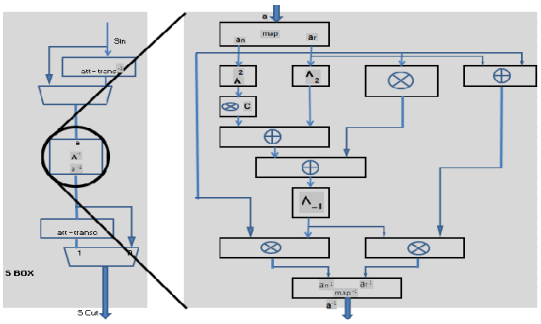


Fig.10.Architecture of AES masked S-Box

The S-Boxes substitute an 8-bit input for an 8-bit output and are based on the arithmetic operations in the finite field GF(2^8).A calculation of this function and its inverse can be done effectively with combinational logic. This approach has advantages over straight-forward implementation using read only memories for table lookups. Most of the functionality is used for encryption and decryption. The Sub Bytes operation is the non-linear component within AES, which makes it particularly difficult to mask. In order to achieve security, use a combination of additive and multiplicative masking. The most tricky part when masking AES is to mask its non-linear operation, which is the finite field inversion (short: inversion) in the S-box, i.e., the Sub Bytes operation.

VI. SIMULATION RESULTS

Based on the modules designed previously, description on logic function of entire system is conducted. Finally, behavioral description on the system was carried out using VHDL hardware description language. In order to verify the correctness of logic functions of the system, simulation of the system with MODELSIM was carried out to verify the correctness of logic functions of this 128-bit mode AES encryption and decryption system. Virtex-6 FPGAs are used to implement the algorithm

Input = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 3707 34

Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

output = 39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

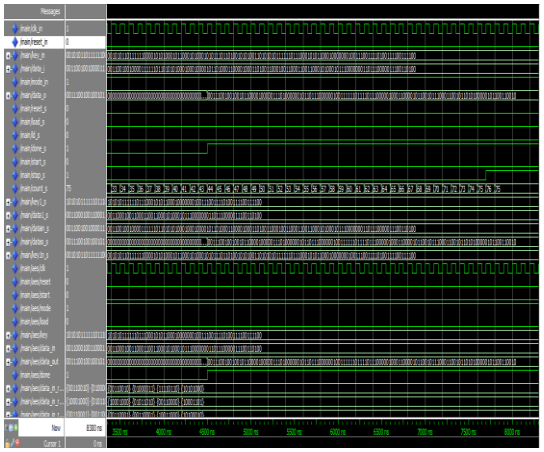


Fig.11.One-Task-One Processor encryption result

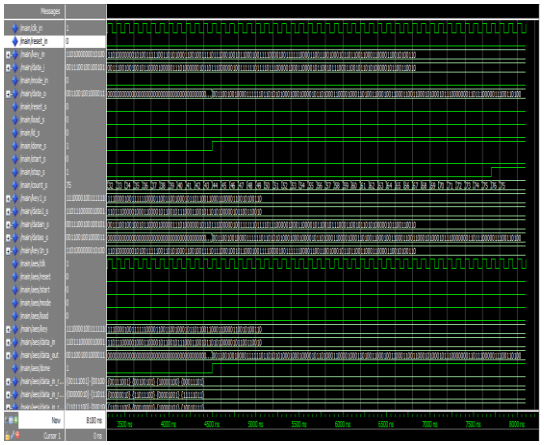


Fig.12.One-Task-One Processor decryption result

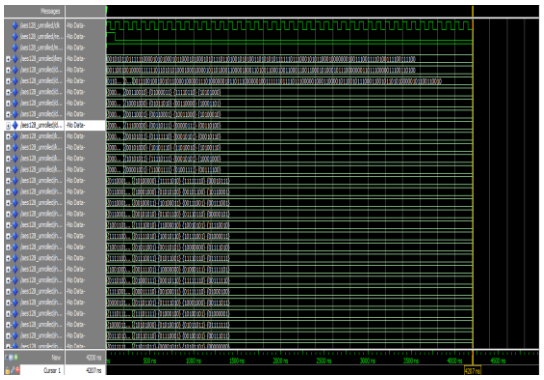


Fig.13.Loop-Unrolled Nine times encryption result

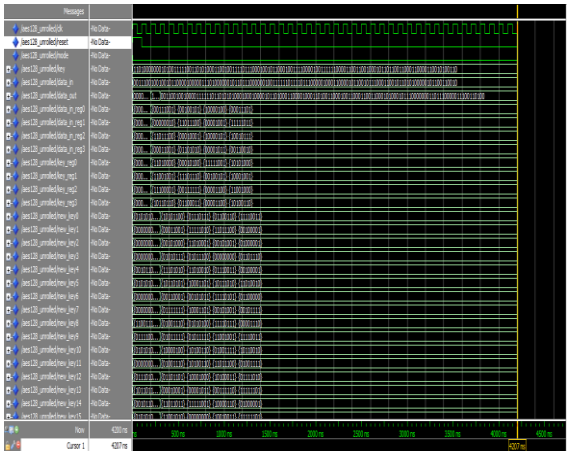


Fig.14.Loop-Unrolled Nine times decryption result

Input = 39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32
Cipher Key = d0 14 f9 a8 c9 ee 25 89 e1 3f 0c c8 b6 63 0c a6
output = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Through the analysis on simulation results of the above test data, under the control of the same key, plaintext is encrypted by the system, and the resulting cipher text is again decrypted by the system, the final resulting data is consistent with the input plaintext data, as well as the test data provided in the reference material. Thus it can be proved that the encryption and decryption work of this system is normal

VII CONCLUSION

High throughput is an important factor for large data transformation systems in SANs. In order to secure "data-at-rest" and enhance the throughput, modern systems shift the encryption procedure from a software platform to a hardware platform.Hardware-based encryption still opens the possibility of DPA and glitch attacks. In this brief, a masked S-box has been proposed to construct the DPA-resistant design with acceptable area on FPGA. The proposed masked S-box only needs to map the plaintext and masking values from GF(2^8) to GF(2^4) once at the beginning of the operation and map the ciphertext back from GF(2^4) to GF(2^8)) once at the end of the operations. The masked S-box has the ability to defend against DPA and glitch attacks,thereby offering high security level. Thus the implementation of masked s-box increases the system security and hence increases the algorithm's performance.

REFERENCES

[1] NIST, "Advanced Encryption Standard (AES)," http://csrc.nist.- gov/publications/fips/fips197/fips-197.pdf, Nov. 2001.
[2] NIST, "Data Encryption Standard (DES)," http://csrc.nist.gov/ publications/fips/fips46-3/fips46-3.pdf, Oct. 1999.
[3] Joan Daemen Vincent Rijmen.AES Proposal:Rijdael
[4] Nicholas Weaver,John Wawizynek. Very High Performance Compact AES Implementations in Xilinx FPGAs
[5] Sounak Samamta.FPGA Implementation Of AES Encryption and Decryption.
[6] Announcing the ADVANCED ENCRYPTION STANDARD(AES).Federal Information Processing Standards Publication 197.November 26,2001
[7] Parallel AES Encryption Engines for Many Core Processor Arrays Bin Liu, Student Member, IEEE, and Bevan M. Baas, Senior Member, IEEE .IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 3, MARCH 2013
[8] High Performance Hardware Implementation of AES Using Minimal Resources Abhijith.P.S, Mallika Srivastava, Aparna Mishra, Dept. of Microelectronics, IIITA, India 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)
[9] A side channel analysis resistant description of the AES S-box.
[10] A systematic evaluation of compact hardware implementation for the Rijndael S-box