DESIGN AND ANALYSIS OF S-BOX DESIGN FOR CRYPTOLOGY

DISSERTATION-II

Submitted in partial fulfillment of the
Requirement for the award of the
Degree of

MASTER OF TECHNOLOGY

By

Sokat Tejani (11610055)

Under the Guidance of
Abhishek Kumar
Assistant Professor, L.P.U

School of Electronics and Electrical Engineering
Lovely Professional University
Phagwara, Punjab
December, 2017

CERTIFICATE

This is to certify that **Sokat Tejani** bearing Registration no.11610055 have completed objective formulation of his Base Paper implementation of the thesis titled, **"Design and Analysis of S-box for Cryptology "** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of thesis has ever been submitted for any other degree at any university.

Abhishek Kumar
Assistant Professor
VLSI Domain
School of Electrical and Electronics Engineering
Lovely Professional University
Phagwara, Punjab

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude and appreciation to my guide **Abhishek Kumar,** for his whole-hearted and invaluable guidance, inspiring discussions, encouragement, and support throughout my work. I found him always sincere in helping me even during his busiest hours of the day. His ardor and earnestness for studies are respected and will never be forgotten. Without his sustained and sincere effort, this report would not have taken this shape.

We are also indebted to all authors of the research papers and books referred to, which have helped us in carrying out the research work.

Sokat Tejani
Reg.No:11610055

DECLARATION

I, **Sokat Tejani**, student of M. Tech VLSI under School of Electronics and Electrical Engineering of Lovely Professional University, Punjab, hereby declare that all the information furnished in this **Dissertation-II** report is based on my own intensive research and is genuine.

This report does not, to the best of our knowledge, contain part of my work which has been submitted for the award of my degree either of this University or any other University without proper citation.

Signature :

Sokat Tejani

Reg.No: 11610055

ABSTRACT

Security in communication has always remained significant concern. With everything becoming data these days, it is essential that no misuse can take place and no harm takes place because of security issues associated with channels. To provide security in insecure environment method such encryption-decryption is vital, cryptography deals with these techniques. Until now, standards such as AES, proved to be robust and to attack those was difficult. Software part robust now, study of crypto-analysis shifted to hardware and side channel analysis came in light for the first time. Side channel, such as power, leaks every detail associated with hardware and modeling of these becomes easy in that case. In cryptographic operations, S-box or substitution is one such module which consumes most of the power and leaks vital power signature to attacker. In this report, in depth focus is given to different S-box architectures and comparative analysis has been done for power, area and timing parameters.

LIST OF ABBREVIATIONS

- **AES : Advanced Encryption Standard**
- **GF : Galois Field**
- **MI : Multiplicative Inverse**

TABLE OF CONTENTS

Title **Page**

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Modern day communication involves many kinds of security threat. Though means of communication have increased significantly and also become sophisticated over the time, security issues were always present. With few attacks, these loop holes came to light and fraternity associated with this modern day science and technology took serious note of that. Over the time, many techniques were developed to address this grey area. Any technique, devised to solve any issue, of communication must confirm three norms of the same which are confidentiality, integrity and availability.

Confidentiality:

Confidentiality of communication said to be maintained, when data which was to be communicated reaches to the desired recipient only and no one except that end user is able to gain any substantial information devising any illicit method. Let say User A wants to communicate with User B over any channel established between compromised here. Instead, even if illicit means if c is not able to know what is being transferred, we can say confidentiality is being maintained. In another way , let say A is sending “hello” to B , if C gets exactly that , confidentiality is compromised here , but if , prior to sending data over channel , it has been changed in some another form , which is mutually acceptable to both A and B , and C has no clue about it , though C gets access to channel , C won’t be able to exactly tell about what is being communicated . So confidentiality is maintained here. This is one of the most fundamental any elementary requirement of any communication that data should reach only to desired recipient and not to anyone else.

Integrity :

When we say that prior to sending data over channel , data is being changed , using some method , which is mutually acceptable to all the parties taking part in communication , it should be taken care that recipient should get exactly what was desired to be communicated and not anything else i.e. data should not change because of such methods. Let say A wants send “hi” to B and for this they use some substitution like,

"tea", so when "tea" gets received by B, he should be able to identify that, "Hi" was conveyed by A. Now C, though gets the message "tea" by manipulating channel, he won't be able to tell what was the purpose of the message and original content of the message. For these, method to convert the data in another form should be known to both participating parties and it should be concealed to any intruder in between. This is the integrity of data over communication. If B is not able to get the original data, integrity of communication is being compromised, can be said easily. So this too is very primary requirement for any communication.

Availability :

For any communication to take place it is must that end users get access to medium which has been devised to establish the communication. What if, few users don't get access to that medium? Communication won't take place at all, and it will result in wastage of all the resources used to make communication possible. So availability of the channel is must. So when A is communicating to B, C should not be denied the access, though what was meant to be for B , should not reach to C . This sum up all the three, Confidentiality, Integrity and avail1bility or CIA. Collectively they make pillars for any communication protocols.

One of the techniques to solve security related threat of data which is being communicated is encryption-decryption. **So** what does it stand for?

Encryption-Decryption:

Data is said to be encrypted, when we get to know true nature of the data, communicated over channel only when we perform some mutually accepted operation on that data. Let say before sending the data over channel, it is being XORed with 1. So on the receiver side, by XORing again with 1 we get to know about the original data (Here, as XORing twice because they cancel each other, i.e. operation to be performed should be opposite of each other). As per technical terminology, operation being performed is known as encryption and operation performed on the receiver side is known as decryption another

input to perform these operations, is known as key; If key gets wrong end user will not be able to get the desired data. So, key is soul of this method.

If Key is known to other intruder or so weak to crack, communication is not of any user then. So any attempt to break or hack the communication is to get to know the key. Algorithm, implementing this should be robust in this case and key should be strong as well, so that trial and error method also does not give any appropriate result. Over the time, many such techniques are developed and situation of 'too many cooks spoil the meal' arose. Search for standard method to do encryption-decryption gave birth to standards such as DES, AES. Also different algorithms came to the picture as well. Two keys method is one such and public key-private key distinction got introduced to crypto-science for the first time. AES, though, is single key method and currently it is used widely.

AES:

The AES is a symmetric key block cipher published by the National Institute of Standards and Technology in 2001. Specifications of the same involve 128 bits of block size and three different key sizes of 128,192 and 256 bits. It is an open algorithm available to the public worldwide.

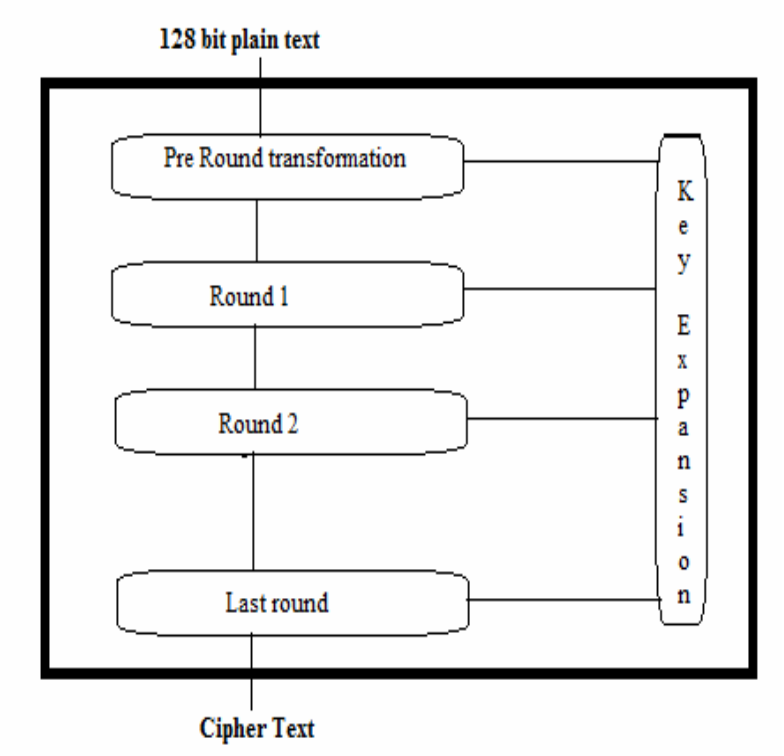


Figure 1 AES Algorithm

AES performs encryption in 10, 12 or 14 rounds depending on key size. It is important to note here that round keys will be always of 128 bits which will be taken care by key expansion algorithm. 128 bits of block data first would be converted into state here in AES. Operation of each round will be performed on each such state. Round in AES (except the last) consists of Sub-Bytes, ShiftRows , MixColumns and AddRoundKey. All these transformations are invertible of course. Last round has only three transformations (All except mixcolumn). Pre round section shown in diagram uses only one transformation (AddRoundKey). At decryption site, the inverse transformations are used : InvSubByte, InvShiftRows , InvMixColumns , AddRoundKey (self invertible).

AES has four types of transformation: Substitution, permutation, mixing, and key adding. Here our entire focus will be on Substitution.

If we analyze AES, we can easily say Substitution or S-box is the most complex or tedious operation out of all operations it involves. Substitution basically is implementation of Shannon's confusion-diffusion principles.

Confusion:

Creating confusion is one of the basic requirements of any cryptographic algorithm. By confusion we mean, relationship between plain text and cipher-text with respect to key is complex one. Let say key is 3 and relationship is that of addition, in this case plain text 5 6 7 8 9 will be converted to 8 9 10 11 12 as algorithm to perform encryption is known (addition in this case) key can be extracted very easily. So in this example there is no confusion, I.e. to make attack impossible, strong confusion property is required in the system.

Diffusion:

Diffusion is element of randomness required to be present. In substitution, for change in one bit of input, substituted output should be change by at least half of bits. This makes it unpredictable and key gets difficult to predict in this case. It limits overall range available for substitution though or makes it difficult to make the table. Substation in AES takes care of the things, confusion and diffusion. Substitution table created using these principles is shown below.

Table 1 Substitution Table

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

Now S-box or Substitution can be implemented using ROM based look-up table method. Here, data which is to be stored, in table format. Data bits, then, will be considered, as no of row and no of column and respective entry of the table will be substituted. This, not only involves, bulky hardware but it also makes overall encryption operation slower and bit lengthy. Not only that, empirical study shows, it is simple to crack. Before talking about cracking AES – look up table algorithm, it is important to know what is the general meaning of cracking of any algorithm and possible ways to make that happen! Algorithm has been cracked, can be said when, key information of that algorithm becomes known to the user. Now using that key, he will get to know what is being communicated and methods to provide security over communication becomes futile. So any attacker's ultimate aim is to get to know the key. Now let say, if key is of 4 bit size, it will not be of any difficult task for attacker to get to know the right key. Here, simply trial and error method will get the right key. This type of attack is known as brute-force attacks and they are termed as weak attacks as key size is very high these days and it is practically impossible to get to know the right key without wasting enormous amount of time on that. This makes sense why this attack is termed as weak attack. Other stronger attacks though are proving to be very potent these days.

AES which uses Look up table can be easily cracked using cache memory concept. Cache is the type of memory which stores frequently used content for speeding up the task in run. So let say encryption algorithm involves two operation, multiplication and addition (decryption will be done using subtraction and division of course) Key for multiplication is say x and addition y which makes ciphered data c= d*x+y where d is the original data or plain text in cryptographic terminology. So limitation of this will be, multiple d will give same c value and it can be known from the access of the cache memory. If cache gets accessed, for different d values, as algorithm for performing encryption-decryption is known, key values can be calculated. So this attack is based on hit and miss of cache memory. Just by observing behavior of cache memory for few input values, key can be extracted. This makes, look-up table based method very weak and prone to attack as it supports cache usage to increase the speed. To prevent this from happening, pre-fetching to cache can be made permanent. But in that case cache loses its

real purpose and that doesn't make implementation any stronger. So, to implementing confusion-diffusion principles, look-up table method utterly failed. Other disadvantages being bulky hardware, slower processing speed.

With significant increase in its usage, demand for efficient method for substitution operation arose which eventually led to development of Galois field method. Galois field method doesn't reveal any information about the data to be substituted. It simply performs mathematical operation on data which comes as an input and performs mathematical operation on that and gives final output of data which is to be substituted.

Substitution using GF method:

Substitution can also be performed using GF field. In this method, the multiplicative inverse of the byte is found in GF(256) with irreducible polynomial(100011011) as the modulus. Inverted byte is then interpreted as a column matrix with the lab at the top and msb at the bottom. This column matrix is multiplied by a constant square matrix **X**, and is added with constant column matrix, **y** to give the new byte. Multiplication and addition done here is GF(2).

Now let us focus on how to find Multiplicative inverse?

First point about MI is they come in pair.

A x B congruence 1 (mod n) than it can be said that A and B are multiplicative inverse of each other.

What is congruence? Congruence is the no which are having same result after modulo operation.

Let say 2 and 12 are two no. If we perform mod 10 operation on both , we will get the same result 2. In this case it can be said they are congruent to each other . For multiplicative inverse pair should be congruent to 1 i.e. simple multiplication of pair and modulo operation performed on that result should give us only 1.

Let us take one example. A=3, B=7, mod 10 than A x B=21 21 mod 10 =1 so yes condition satisfied.

3 and 7 are multiplicative inverse for Z10; similarly can be calculated for others as well.

In nutshell, A has multiplicative inverse in Zn if and only if gcd(n,a)=1 i.e. if they are relatively prime.

Now after multiplicative inverse we are supposed to perform column matrix multiplication by converting inverted result in column matrix as discussed earlier. At last stage we are required to perform addition these two operations are quite similar to affine cipher operation so let us discuss that.

Affine Transformation:

Affine transformation is combination of multiplicative and additive transformation. To state this mathematically,

Cipher text = { (Plain text x K1) + K2 } mod 26

Plain text = { (Cipher text – K2) x (multiplicative inverse of K1) } mod 26

Eg : hello with key pair of (7,2)

H -> 07 encryption ({07x7)+2} mod 26 -> 25 -> Z

E -> 04 encryption {(04x07)+2} mod 2 -> 04 -> E

And so on… Result would be ZEBBW.

Here in Substitution, keys would be column matrices.

Result of this operation will be used to substitute the data and hence this is how substitution is performed using mathematical operation.

If we compare look up table method and Galois method, one obvious conclusion will be , power taken by the operations , involved with Galois method, is on higher side for GF method than look up table method . Reason for this fairly simple one, in look up table method only scanning i.e comparing is required while in GF method as stated earlier so many mathematical operation is needed to be performed to get the “to be substituted ” data as a result.

With advancement in the cryptanalysis, newer method to attack AES came to light. Algorithm, being out in public domain, assures security as far as key is concerned. It doesn’t leak, in any possible way, any information related with key. It is not possible to reverse engineer this algorithm and brute force attack is way too much time consuming

and involves computation beyond competence. So with time, methods were developed to attack hardware and by making model of the same, techniques were developed to predict the key values out of that model. Information, used to make the model should represent the hardware as accurately as possible and as many details as possible should be used to make such a method. Machine learning algorithm to be used, should also be fast and accurate.

Method which got prominence to attack the hardware is differential power analysis. It comes under side channel analysis. How it can be prevented is the area under research in my thesis but first it is important to know how it is devised to attack the system.

Side Channel Analysis:

Side channel is an attack when information to crack the system is derived from the hardware used for implementation of the cryptographic algorithm. Recently, it has come to know that this form of the attack is the strongest and really no satisfactory guarding mechanism is yet to be developed against this form of attack. Not that it takes advantage of loop holes remained in the hardware, however strong implementation is , this attack just tries to emulate the actual hardware and efficient modeling techniques takes care of the remaining things .

To make the model of any system, it is very important that we have exact and minute details of system's behavior for majority of the possible inputs. Once this has been studied, learning can be used to create the model. Suitable learning method and algorithm can be used. Test patterns will let attacker know, whether he is successful in making the model or not. Once model is ready, system can be cracked for the key.

So, what are the features, which can be used to make the replica? Only requirement here is , any parameter to be the feature of the system is , it should give exact idea , how system is reacting to input given and how system's response changing with respect to change in inputs. Power , timing delay , power-delay product , sound generated by system , radiation it is generating , or any such property and manipulation form of them can be

used to train the system. So to avoid such attacks it is important to make system's behavior unpredictable in some sense. There should be some randomness but that is not the case with power analysis. Difference in power consumption in true sense, leaking the system's response and helping to create the model of the same. To avoid that it is desired that, power consumption becomes uniform and no power pattern gets generated from the system. It is difficult to design system that way though. How attack can take place is explained briefly below.

Crude explanation of SCA attack is attempted here. First there are few assumptions which are always true when any attempt to break the security is in the go. First of these is , attacker has full knowledge of hardware. Second, algorithm , performing the operation is known to him as well. Now with these in mind, let say we are trying to break the system and extract the key out of it and for that we are using machine learning methods. Let say key is K and in the algorithm it interacts with plain text messages. Now we are recording the behavior of hardware for different plain texts and can exactly know how it is responding to them. This leaves loop hole for developing software system which attempts parody of the actual hardware and then we can , comparing the predicted key's response for particular P with actual hardware's response for the same P , K can be extracted . To make this possible machine learning is required, which uses learning. Here learning is in terms of response of the hardware say power it consumes, delay it produces, and sound it makes or in some cases radiation it generates. Accuracy of these methods has always been remain constrain though if required amount data is available, replica of the system will become easy job. Responses which are used to make the model are known as features and power of the system is one of the most important one.

Chapter 2

Scope of the Study

S-box design is one of the most critical aspect of AES cryptographic algorithm. Conventional designs are leaking most of the power and hence falling victim to differential power analysis and that way side channel analysis. That way , it is important to study conventional design and make it robust and make it such a way that it doesn't leak any information which can be used to model that system. Entire focus of this study , that way is on S-box, cracking , re-designing which is the aim of study.

Chapter 3

Objective of the Study

Objective of the study is strictly limited to improving the S-box design. Differential Power Analysis is proving to be one of the strongest attacks and it is strong enough to crack AES. Objective of the study hence is to make S-box design such a way that it consumes uniform power irrespective of the input patterns. I.e. it should not produce any power signature which can be used to model the hardware.

Chapter 4

Literature Review

This part conveys the survey of writing about the present input operational amplifier with different strategies engaged with it. I had knowledge look on a few journal and conference papers, identified with the desired S-Box for AES. I have customized a portion of the journal and conference papers organized an outline on those.

Design of Low Power S-box architecture Level using GF

In this paper authors proposed high level transformation technique on S-box. Concept here is by introducing pipelining and keeping the input at same frequency power consumption can be reduced. Proposed architecture and conventional architecture have been given here in block diagram format. [1]

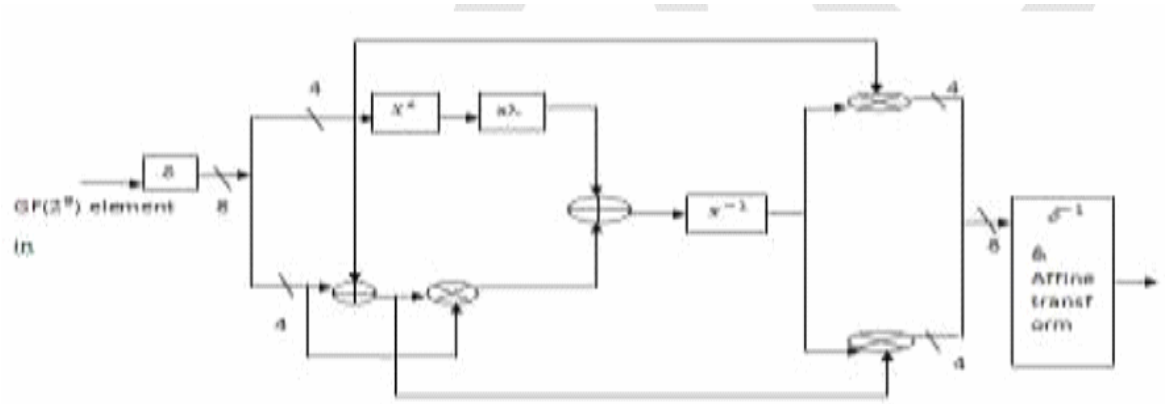


Figure 2 Convention S-box using GF

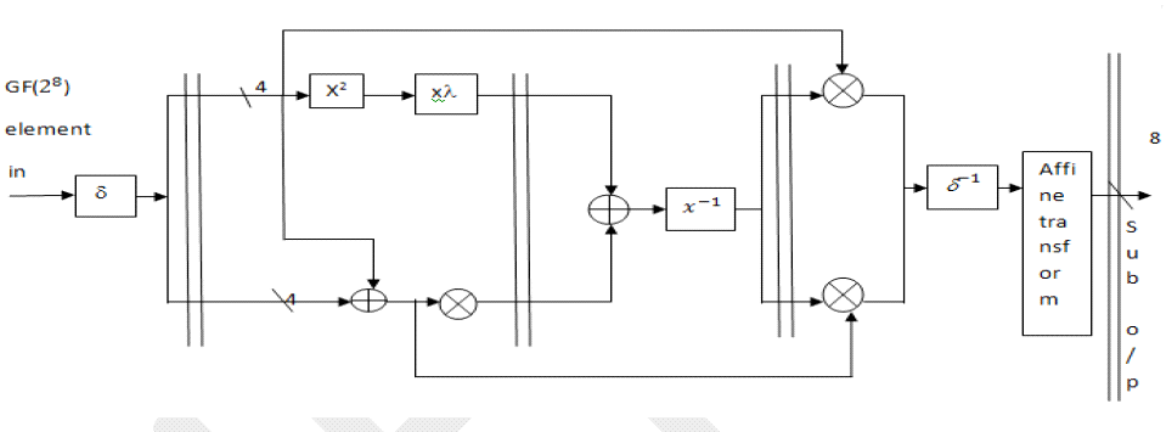


Figure 3 Pipelined architecture

Low Latency VLSI architecture of S-box for AES encryption

In this paper authors came up with innovative method to find multiplicative inverse which is faster in compare to conventional method which ultimately leads to faster substitution. Proposed architecture and conventional block diagrams have been shown here. [2]

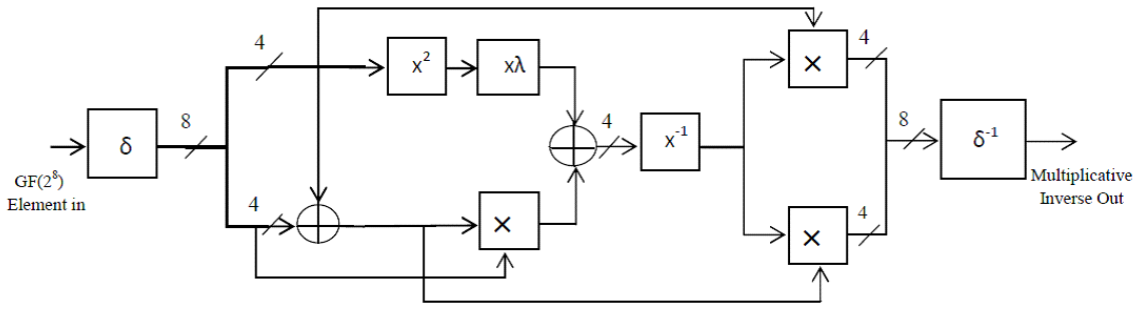


Figure 4 Conventional method to obtain MI

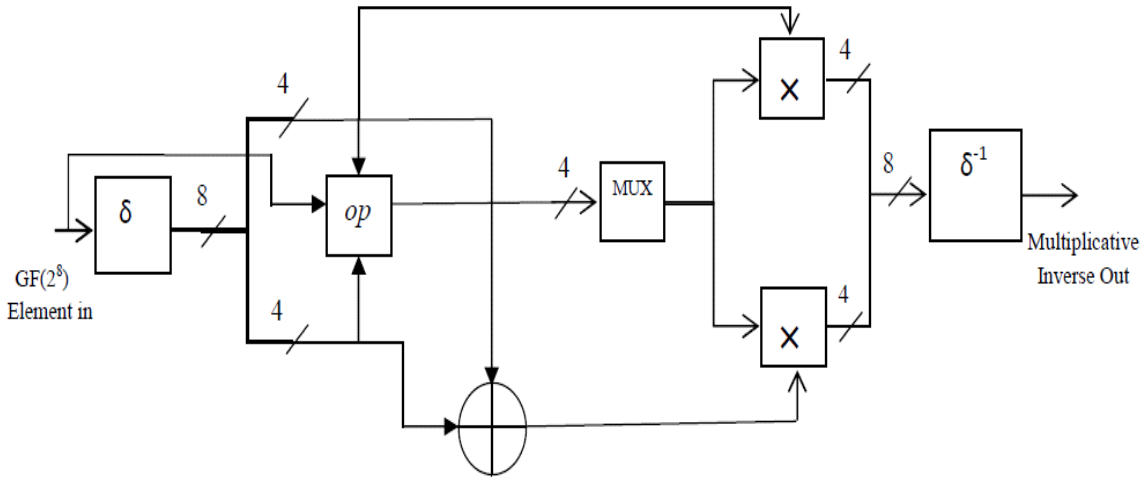


Figure 5 MI Circuit Diagram

Side channel analysis resistant description of the AES S box

One of the counter measures to avoid side channels is masking. Here in this paper authors propose additive and multiplicative masking for inversion operation involved in substitution. The novel feature here is proposed method makes inversion a linear operation (on GF(4)) and hence masking can be done easily. Ultimately this is also provides security against zero-value attack. [3]

A VLSI design flow for secure side channel attack resistant ICs

In this paper , authors proposed some key modifications to make ICs side channel attack resistant. The same has been mentioned below. Empirical analysis suggests that, ICs made from this flow do not reveal key even after more than 2000 Differential power analysis records while simple ICs will divulge that in just 200 readings.[4]

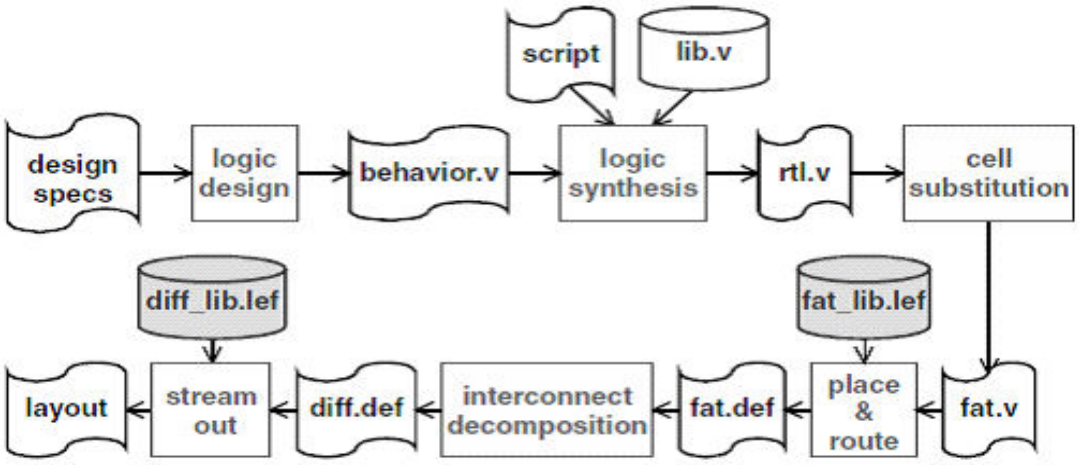


Figure 6 Secured Design flow

Side channel attacks: Ten years after its publication and the impacts on cryptographic module security testing

In this paper, author gives idea about what side channels are how they can lead to attack and what type of side channel plays role in what type of attack. Author describes different types of side channels and gives glimpses on those points. Classification of attack is also one noteworthy thing presented in paper. [5]

Chapter 5
Research Methodology

Method 1: Conventional Look-Up table based S-box

Traditional look up table involves ROM implementation. Memory will be used to hold the values and scanning will be done across rows and columns to fetch the desired value. Look up table is shown in fig below. Operation is pretty simple here. 8 bits data which is to be substituted will be used to get the appropriate "to be substituted" value from the table. For this, 8 bits data will be considered as combination of two nibbles. First of which, will give the number of row from where data is to be fetched and second of which will give the number of column from where data is to be fetched. Intersection point of these two entries will give the data to be substituted. On the receiver side similar table will be there and same method will be used there as well to get the desired outcome. From implementation point of view, this method is fairly simple and right to say that it doesn't involve any complexity except for the bulky memory which will be required to implement this.

Verilog code was implemented for the same and RTL analysis performed for the same code shows details of area , power , and timing details associated across 3 technologies (45nm , 90 nm and 180 nm for slow and fast libraries) Results of the same are given below.

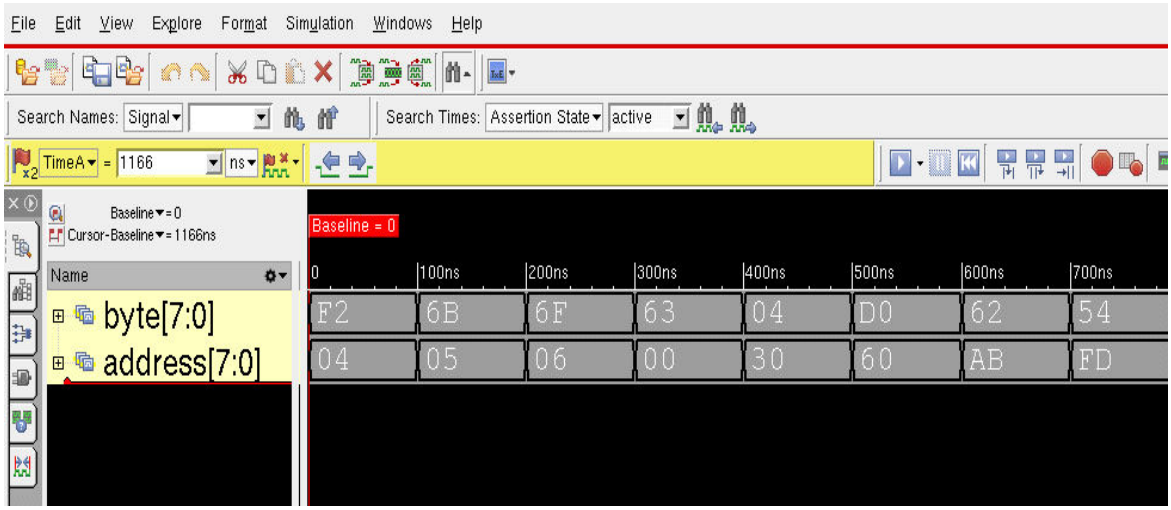


Figure 7 Output of look-up table based design

Table 2 RTL results analysis

		45nm		90nm		180nm	
		Slow	Fast	Slow	Fast	Slow	Fast
Power(nw)	Dynamic	22312.51	40715.29	39308.08	56032.39	131495.3	220603.2
	Static Leakage	37.61	135.41	5670.21	14214.73	124.59	54.75
Area	Cells	460	443	411	409	418	414
	Inverters	97	80	39	37	44	43
	Logic	363	363	372	372	371	371
Worst Path Delay(ps)		3319.5	1035.3	3070.1	809.3	4477.1	1702.2

Observation:

Power for same technology, slower library consumes less power than faster one, worst case delay is less for faster library than slower library which is either because of removal of few logic blocks or inverters and overall optimization of the design. Improved performance from 180 to 90 to 45 in case of area power and delay. Leakage power is higher in case of 90nm technology though. Area is least in case of fast 45nm technology while most in case of 90nm technology.

Method 2: Modified Look-Up table based S-box using decoders and multiplexer

To make the ROM based implementation faster, decoder-multiplexer based technique is used. Here 8 bits data which was divided in nibbles gets further divided which is explained below. Based on first two bits of the 8 bits data, group of the "to be substituted data" will be decided. To make it faster, decoders are used. [6]

Here, in this method, nibbles used to decide row and column entry are further divided. First two bits of first nibble are used to decide the group and accordingly four groups will be created. Second two bits of first nibble will be used to select the row entry. First two bits of second nibble will be used to select column entry. This will result in four entries as possible output. Mux will be used and as per last two bits, one of the 4 entries will be selected. It is shown graphically below.

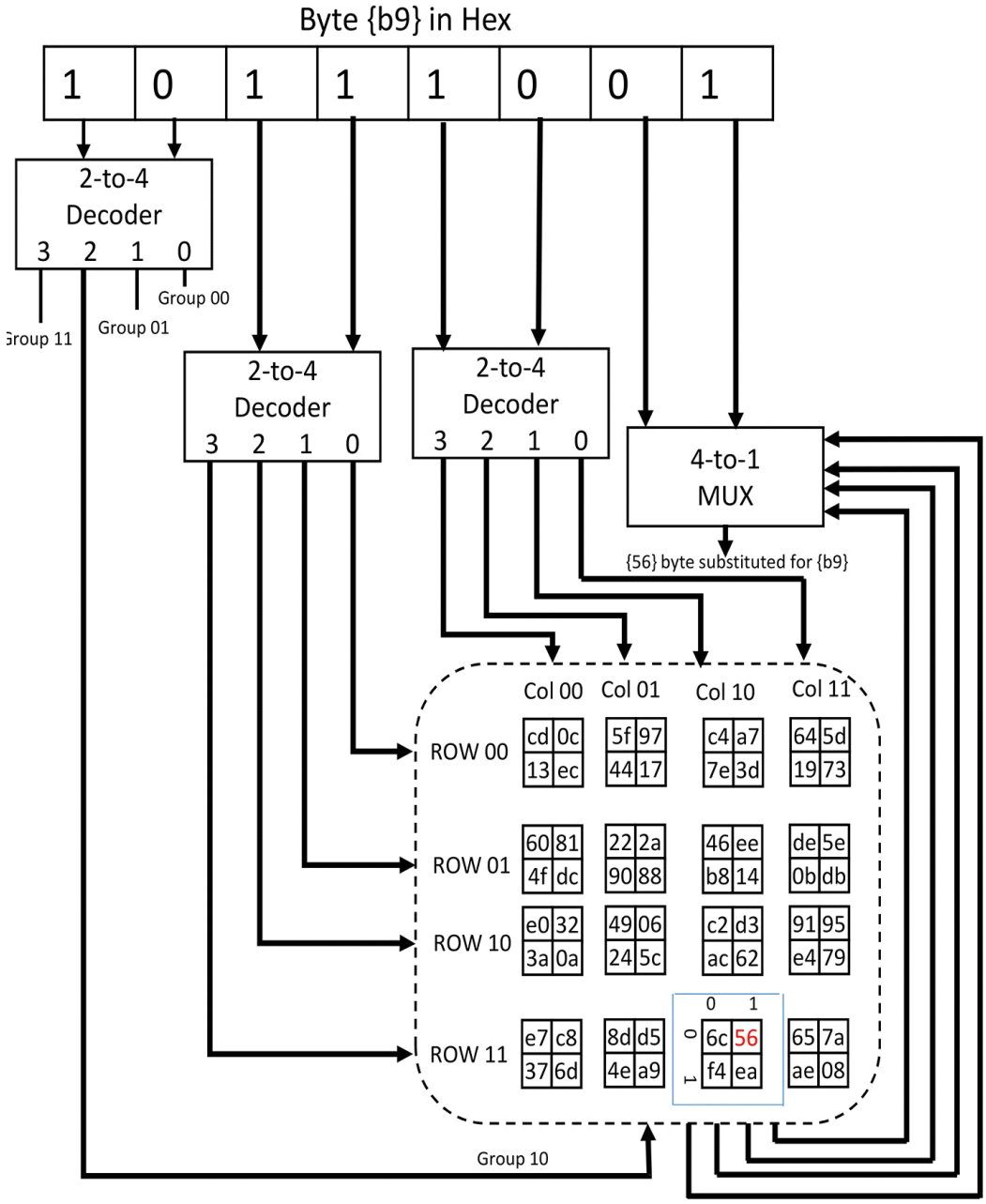


Figure 8 Addressing decoding method for modified look up table

Groups which are created and new substitution table are given below. Verilog code was implemented to give effect to this mechanism, output and RTL analysis of the same is given below in the tabular format.

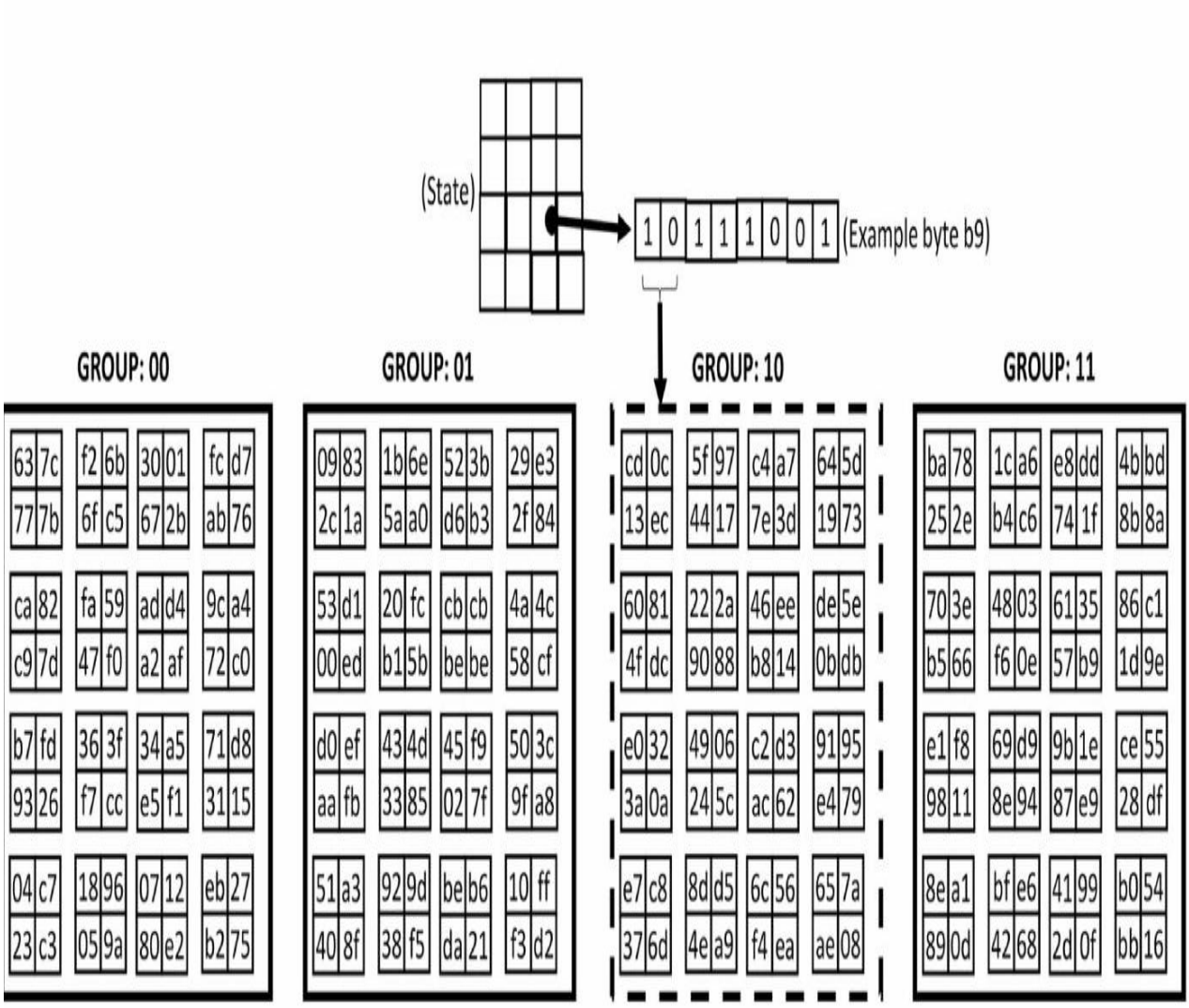


Figure 9 Groups

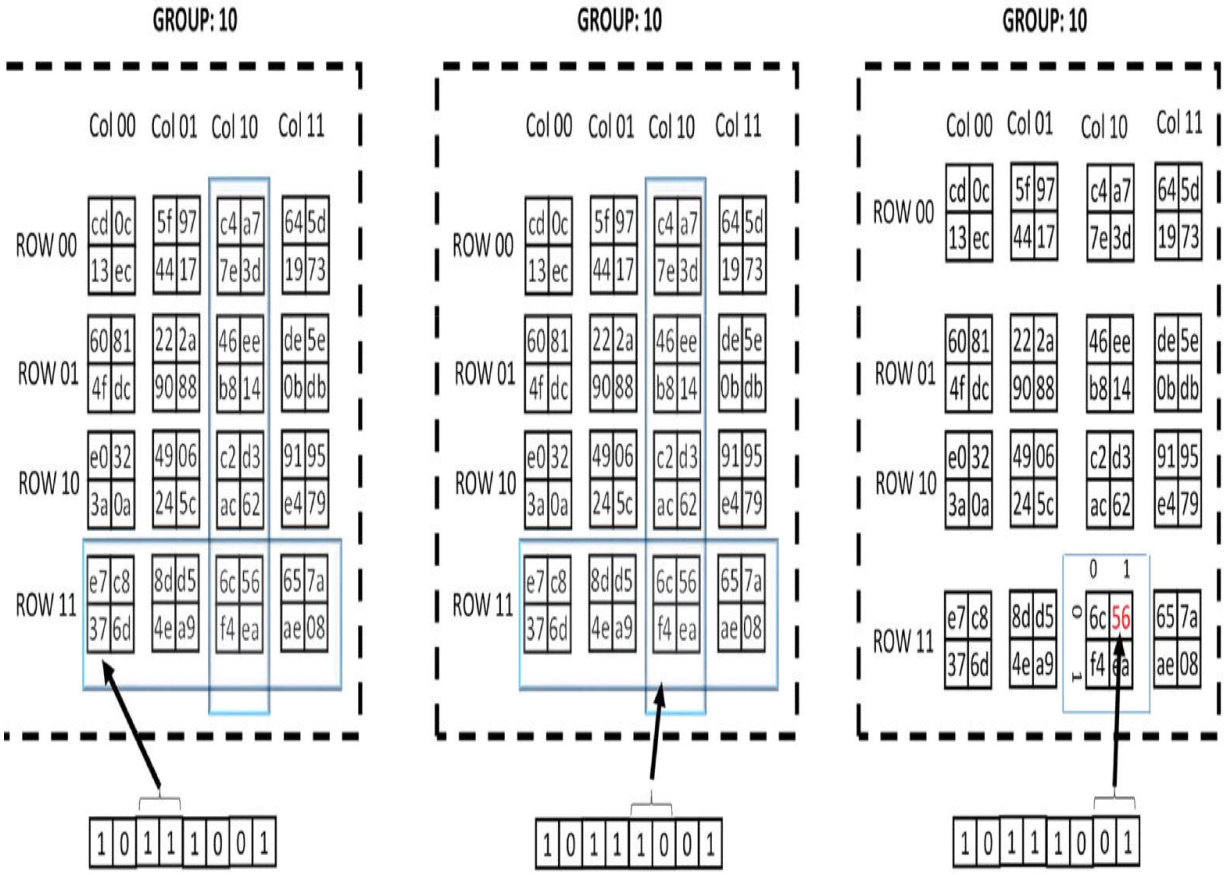


Figure 10 Decoding patterns

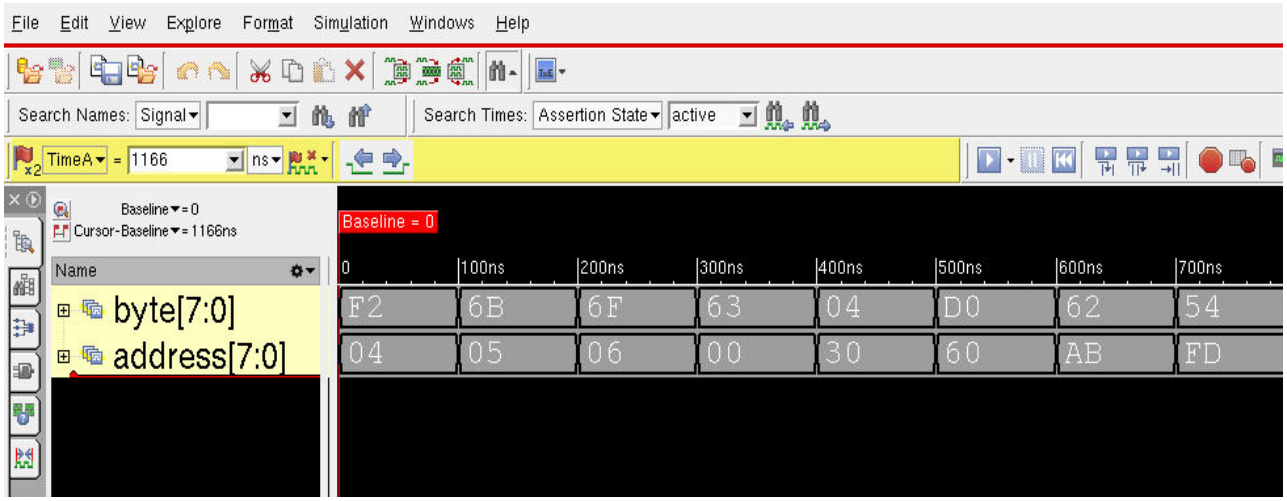


Figure 11 Modified look up table based output

Table 3 RTL results

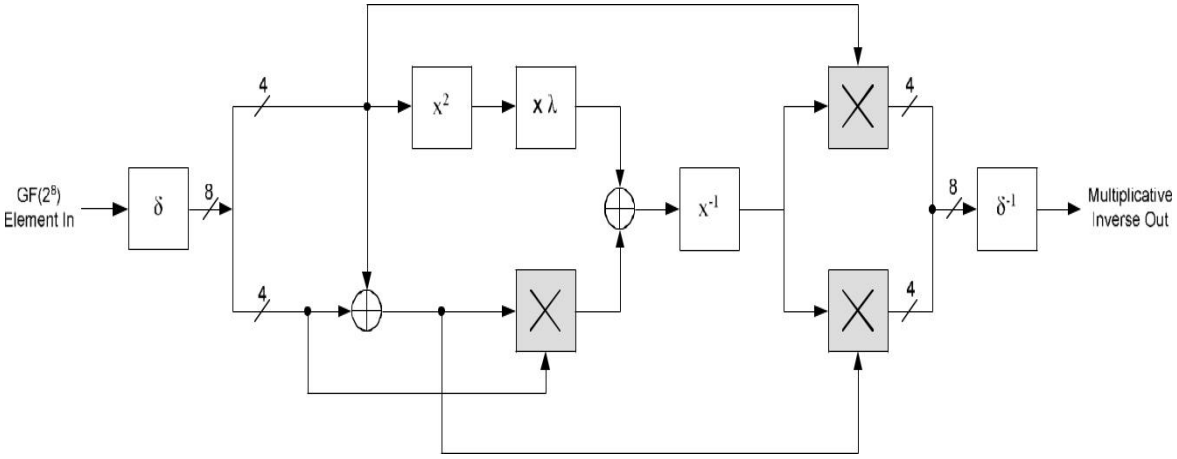
		45nm		90nm		180nm	
		Slow	Fast	Slow	Fast	Slow	Fast
Power(nW)	Dynamic	28590.55	43059.27	49448.73	73248.26	208492.9	367766.1
	Static Leakage	27.48	91.28	4955.27	12384.88	111.21	43.16
Area	Cells	437	428	405	405	410	411
	Inverters	114	105	63	63	68	69
	Logic	323	323	342	342	342	342
Worst Path Delay(ps)		1845.1	624.7	2300.8	727.7	3961	1757.4

Observation:

For same technology, faster library, by reducing few redundant cells, improves upon worst path delay. Price for this, though, paid in terms of power. Faster library takes more power than slower one for same technology. Similar statement can be made about Static power. As technology changes, performance gets improved as we move from 180 to 90 to 45nm. Area is increasing, which is classic area v/s speed v/s power enigma.

Method 3 : S-box using Galois field

Substitution using GF method basically involves two operations. One is multiplicative inverse and the other is affine transformation. It is not possible to find the MI in gf(256) straight away. First it will be is-formed to lower GF fields to gf(16) and operations as given below in the chart will lead to MI output. How this arrived is explain in brief in following paragraphs.[7]



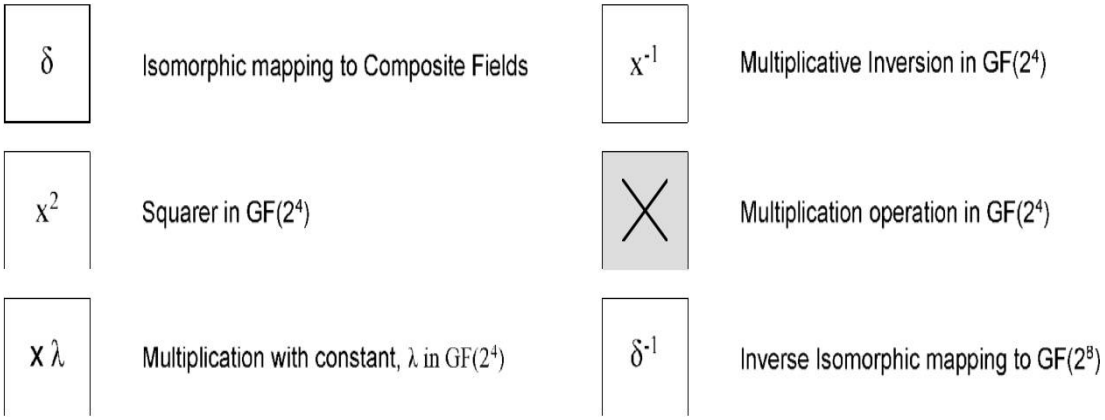


Figure 12 Method for obtaining MI

Following multiplicative inverse module, affine transformation module will be implemented. Affine transformation involves two operations, multiplication and addition. Matrix multiplication, too , will turn in addition operation and in turn simple XORing

will be required. Following multiplication, another Xoring will be performed for addition operation. Combining both Substitution output will be achieved. Multiplicative inverse circuit is mentioned above with each and every component details. This, along with affine module, was implemented in verilog language on NCSIM tool. The output and analysis report in tabular format given below.

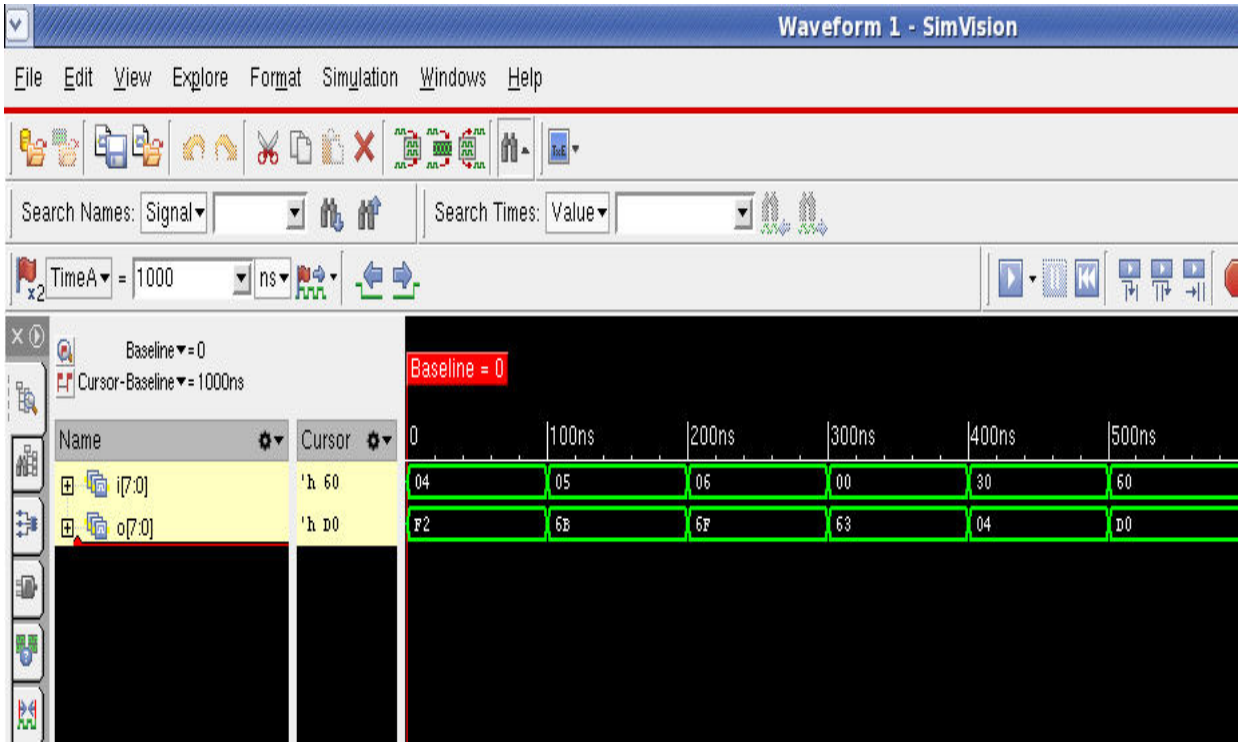


Figure 13 GF based S-box Output

Table 4 RTL result

		45nm		90nm		180nm	
		Slow	Fast	Slow	Fast	Slow	Fast
Power(nW)	Dynamic	55749.18	85664.85	75096.6	119434.2	312859.5	517037
	Static Leakage	28.832	77.817	7169.91	13600.86	138.42	119.49
Area	Cells	170	165	182	182	170	170
	Inverters	27	22	13	13	5	5
	Logic	143	143	169	169	165	165
Worst Path Delay(ps)		in pin 2 to outpin 4 4710	in pin 2-outpin 4 1440	In2-out 2 5787.6	inp2-out 5 1567.8	In7-out2 9205.1	in7-out0 3656.60

Observation:

For same technology, faster library, by reducing few redundant cells, improves upon worst path delay. Price for this, though, paid in terms of power. Faster library takes more power than slower one for same technology. Similar statement can be made about Static power. As technology changes, performance gets improved as we move from 180 to 90 to 45nm. Area is increasing, which is classic area v/s speed v/s power enigma.

Method 4: Modified GF based S-box using look-up table

Blend of look up table method and GF method. It can be concluded that, In GF method, almost all the power is consumed by MI module. If we find the MI values and make table of the same, power reduction can be achieved in the GF based substitution method[8].

Here, trick is, both the operation, MI and affine transformation will be performed but for MI module, look-up table will be used. Following that affine will be same as in the case of earlier method. Output of substitution operation will remain same and that is attached below. Power, area, timing analysis report in tabular format.
[To make this implementation faster we can use decoder-mux method as implemented in method 2. It will be done at later stage.]

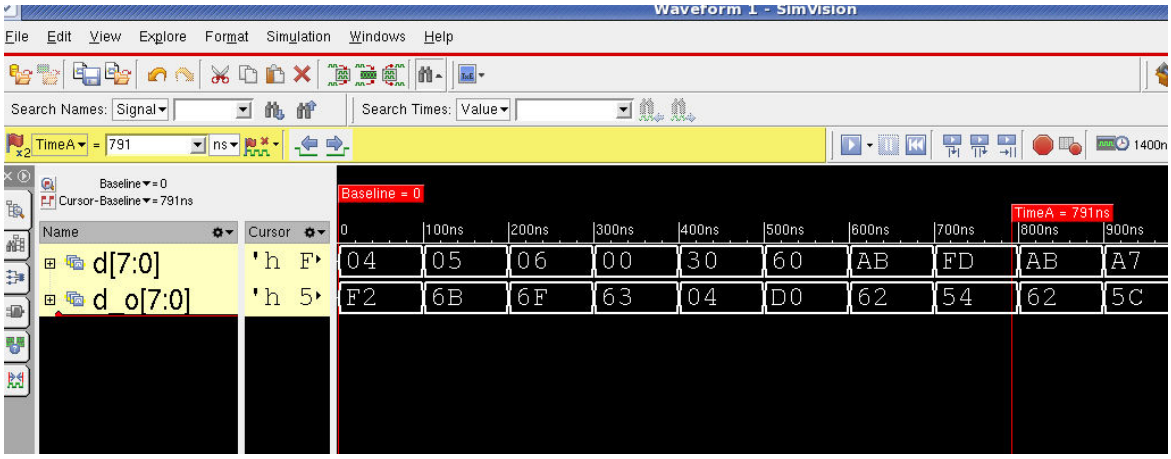


Figure 14 Modified GF method output

Table 5 RTL result

		45nm		90nm		180nm	
		Slow	Fast	Slow	Fast	Slow	Fast
Power(nW)	Dynamic	32373.08	54428.21	56383.37	83858.05	224814.83	373557.54
	Static Leakage	52.56	172.198	7188.96	17721.43	150.884	56.913
Area	Cells	508	509	526	526	515	515
Worst Path Delay(ps)		4468	1426	3312	883	4462	1840

Observation:

For same technology, faster library, by reducing few redundant cells or by utilizing other methods, improves upon worst path delay. Price for this, though, paid in terms of power. Faster library takes more power than slower one for same technology. Similar statement can be made about Static power. As technology changes, performance gets improved as we move from 180 to 90 to 45nm. Area is increasing, which is classic area v/s speed v/s power enigma.

Chapter 6

Expected Outcomes

To make cryptographic hardware robust and attack proof it is important to know how it can be attacked . To attack the system which involves any cryptographic implementation , first step is to make mathematical replica of the hardware. For that it is important to study the important features of the system when it is under operation. Feature , selected , to make the model , should be such that it represents entire circuit as a whole . Not only that , variation in inputs should be replicated appropriately in that feature value. Also it should be easy to use as training data .

With these things in mind , many of the feature associated with the system fails to make the cut but few make to that list . Power , EM radiation , Sound such signatures can be used to train the model to replicate the hardware. As it is not possible to get the power patterns associated with system for particular inputs in the digital implementation , it is crucial to make actual hardware circuitry in the availabe tool . In my case it will be Virtuoso .

Sufficient amount of input samples of 128 bits , with fixed key of 128 bits , power associated with the circuit will be recorded. Data cannot be randomly taken as no of 1's and 0's present in the input pattern make significant change in response of the system. Every possible combination is not possible to ocover as input is that of 128 bits but this 2 to power 128 combinations will be divided across few classes and from those classes sufficient input samples will be used to make the model. Once model is ready , it will be tested for its accuracy . If it satisfies that bottle neck , we are ready to launch attack on the cryptographic system.

Method to be implemented for attack is differential power analysis. First of all random key will be selected and input will be given . Now from that key one particular bit will be selected. For different inputs key will be kept 1 and 0 and hence two iterations will be

performed for same set of input . Power will be recorded . If the difference is near to zero , predicted bit is right (1) and if difference is high prediction is wrong . This will be done repeatatively to extract the entire key.

Once it is proven that AES can be cracked this way , attempts to make it robust to these attacks will be made . Fewer techniques , to this effect , have been thought upon and attempted . To make hardware DPA-proof , it is important that hardware doesn't generate any power signature with respect to input. Hardware should be giving the same power irrespective input given to it . Adding dummy operation , adding noise are few of the techniques to make this happen. Dummy operations should not be such that it changes any operation of the system .

With these things implemented , it can be expected that model will help first cracking of the AES algorithm and at later stage it can be studied what and where improvement is required and of which nature . Future work also involves making GDSII of the implemented codes. For this Encounter tool will be used

Chapter 7

Work Plan with Timelines

Time	Work
August,2017	Literature Review for selecting base paper
September,2017	Two papers were implemented
October,2017	Two more papers with different techniques were implemented
November , 2017	Analysis and Report work

Chapter 8

Results and Discussion

To gain the significant insight, RTL results are compared over different libraries and different technologies. Conclusions drawn from these comparisons are pretty much interesting.

1. Dynamic power comparison over slow library for 45nm over different methods

Table 6 Dynamic power for 45nm (slow)

Look up	modified look up	Gf	Modified gf
22312.51	28590.55	55749.18	32373.08

It can be seen from this, that because of multiplicative inverse operation, GF method consumes highest power. When, to find MI, look-up table is used, it takes less power. Look up table method takes least power, and Modified look up table method increases speed, but power is on higher side in that case.

2. Dynamic power comparison over slow library for 90nm over different methods

Table 7 Dynamic power for 90nm (slow)

Look up	modified look up	Gf	modified gf
39308.08	49448.73	75096.6	56383.37

It can be seen from this, that because of multiplicative inverse operation, GF method consumes highest power. When, to find MI, look-up table is used, it takes less power. Look up table method takes least power, and Modified look up table method increases

speed, but power is on higher side in that case.

3. Dynamic power comparison over slow library for 180nm over different methods

Table 8 Dynamic power for 180nm (slow)

look up	modified look up	Gf	Modified gf
131495.3	208492.91	312859.51	224814.8

It can be seen from this, that because of multiplicative inverse operation, GF method consumes highest power. When, to find MI, look-up table is used, it takes less power. Look up table method takes least power, and Modified look up table method increases speed , but power is on higher side in that case .

4. Static power comparison over slow library for 45nm over different methods

Table 9 Static power for 45nm (slow)

look up	modified look up	Gf	modified gf
37.61	27.48	28.832	52.56

Static power has more to do with area or no of cells in the configuration. Not every time this relation can be established though . But it is obvious that higher no of components present in the circuit will attract higher leakage than circuit which has less components. Also , slower library has low switching activities than faster library so just like dynamic power , leakage will also be less in case of slow library for same technology.

5. Static power comparison over slow library for 90nm over different methods

Table 10 Static power for 90nm (slow)

look up	modified look up	Gf	modified gf
5670.21	4955.27	7169.91	7188.96

Static power has more to do with area or no of cells in the configuration. Not every time this relation can be established though. But it is obvious that higher no of components present in the circuit will attract higher leakage than circuit which has less components. Also, slower library has low switching activities than faster library so just like dynamic power , leakage will also be less in case of slow library for same technology.

6. Static power comparison over slow library for 180nm over different methods

Table 11 Static power for 180nm (slow)

look up	modified look up	gf	modified gf
124.59	111.21	138.42	150.884

Static power has more to do with area or no of cells in the configuration. Not every time this relation can be established though . But it is obvious that higher no of components present in the circuit will attract higher leakage than circuit which has less components. Also , slower library has low switching activities than faster library so just like dynamic power , leakage will also be less in case of slow library for same technology.

7. Timing 45nm slow

Table 12 Delay for 45nm (slow)

look up	Modified look up	gf	modified gf
3319.5	1845.1	4710	4468

Though GF method reduces area significantly, scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF method improves upon that but it can never match what Modified look-up table method has to offer as far as speed is concerned.

8. Timing 90nm slow

Table 13 Delay for 90nm (slow)

look up	modified look up	gf	modified gf
3070.1	2300.8	5787.6	3312

Though GF method reduces area significantly, scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF method improves upon that but it can never match what Modified look-up table method as to offer as far as speed is concerned.

9. Timing 180nm slow

Table 14 Delay for 180nm (slow)

look up	Modified look up	Gf	modified gf
4477.1	3961	9205	4462

Though GF method reduces area significantly, scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF method improves upon that but it can never match what Modified look-up table method has to offer as far as speed is concerned.

10. Area 45nm slow

Table 15 Area for 45nm (slow)

look up	Modified look up	gf	Modified gf
460	437	170	508

Just see the reduction in total no of cells when we implement substitution using GF method! This method significantly reduces bulk components. Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also , as modified GF method has both elements , look-up table and components for affine transformation it takes highest no of cells than any other technique.

11. Area 90nm slow

Table 16 Area for 90nm (slow)

look up	Modified look up	gf	Modified gf
411	405	182	526

Just see the reduction in total no of cells when we implement substitution using GF method! This method significantly reduces bulk components. Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also, as modified GF method has both elements, look-up table and components for affine transformation it takes highest no of cells than any other technique.

12. Area 180nm slow

Table 17 Area for 180nm (slow)

look up	modified look up	gf	modified gf
418	410	170	515

Just see the reduction in total no of cells when we implement substitution using GF method! This method significantly reduces bulk components. Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also, as modified GF method has both elements, look-up table and components for affine transformation it takes highest no of cells than any other technique.

13. Dynamic power comparison over fast library for 45nm over different methods

Table 18 Dynamic Power for 45nm (fast)

look up	modified look up	Gf	Modified gf
40715.29	43059.27	85664.85	54428.21

It can be seen from this, that because of multiplicative inverse operation, GF method consumes highest power. When, to find MI , look-up table is used , it takes less power. Look up table method takes least power, and Modified look up table method increases speed , but power is on higher side in that case .

14. Dynamic power comparison over fast library for 90nm over different methods

Table 19 Dynamic Power for 90nm (fast)

look up	modified look up	Gf	Modified gf
56032.39	73248.26	119434.2	83858.05

It can be seen from this , that because of multiplicative inverse operation , GF method consumes highest power . When to find MI look-up table is used, it takes less power. Look up table method takes least power and Modified look up table method increases speed, but power is on higher side in that case.

15. Dynamic power comparison over fast library for 180nm over different methods

Table 20 Dynamic Power for 180nm (fast)

look up	Modified look up	gf	modified gf
220603.21	367766.08	517037	373557.5

It can be seen from this , that because of multiplicative inverse operation , GF method consumes highest power . When, to find MI , look-up table is used , it takes less power. Look up table method takes least power, and Modified look up table method increases speed , but power is on higher side in that case .

16. Static power comparison over fast library for 45nm over different methods

Table 21 Static Power for 45nm (fast)

look up	modified look up	gf	modified gf
135.41	91.28	77.817	172.198

It can be seen from this , that because of multiplicative inverse operation , GF method consumes highest power . When , to find MI , look-up table is used , it takes less power. Look up table method takes least power , and Modified look up table method increases speed , but power is on higher side in that case .

17. Static power comparison over fast library for 90nm over different methods

Table 22 Dynamic Power for 90nm (fast)

look up	Modified look up	gf	modified gf
14214.73	12384.88	13600.86	17721.43

It can be seen from this , that because of multiplicative inverse operation , GF method consumes highest power . When , to find MI , look-up table is used , it takes less power. Look up table method takes least power , and Modified look up table method increases speed , but power is on higher side in that case .

18. Static power comparision over fast library for 45nm over different methods

Table 23 Static Power for 45nm (fast)

look up	modified look up	gf	modified gf
54.75	43.16	119.49	56.913

It can be seen from this , that because of multiplicative inverse operation , GF method consumes highest power . When , to find MI , look-up table is used , it takes less power. Look up table method takes least power , and Modified look up table method increases speed , but power is on higher side in that case .

19. Timing 45nm fast

Table 24 Timing for 45nm (fast)

look up	Modified look up	gf	modified gf
1035.3	624.7	1440	1426

Though GF method reduces area significantly , scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF

method improves upon that but it can never match what Modified look-up table method has to offer as far as speed is concerned .

20. Timing 90nm fast

Table 25 Timing for 90nm (fast)

look up	modified look up	gf	modified gf
809.3	727.7	1567.8	883

Though GF method reduces area significantly, scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF method improves upon that but it can never match what Modified look-up table method has to offer as far as speed is concerned.

21. Timing 180nm fast

Table 26 Timing for 180nm (fast)

look up	modified look up	Gf	modified gf
1757.4	1702.2	3656.6	1840

Though GF method reduces area significantly, scanning through look-up table is way much faster than performing mathematical operation to get the substitution. Modified GF method improves upon that but it can never match what Modified look-up table method has to offer as far as speed is concerned.

22. *Area 45nm fast*

Table 27 Area for 45nm (fast)

look up	Modified look up	gf	modified gf
443	428	165	509

Just see the reduction in total no of cells when we implement substitution using GF method! This method significantly reduces bulk components. Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also, as modified GF method has both elements , look-up table and components for affine transformation it takes highest no of cells than any other technique.

23. *Area 90nm fast*

Table 28 Area for 90nm (fast)

look up	modified look up	gf	modified gf
409	405	182	526

Just see the reduction in total no of cells when we implement substitution using GF method ! This method significantly reduces bulk components . Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also , as modified GF method has both elements , look-up table and components for affine transformation it takes highest no of cells than any other technique.

24. *Area 180nm fast*

Table 29 Area for 180nm (fast)

look up	modified look up	gf	modified gf
414	411	170	515

Just see the reduction in total no of cells when we implement substitution using GF method ! This method significantly reduces bulk components . Price is paid in terms of power and timing but it provides security than look-up table method and makes hardware simple than other techniques. Also , as modified GF method has both elements , look-up table and components for affine transformation it takes highest no of cells than any other technique.

Chapter 9

Summary and Conclusions

S-box using different methods was implemented and rigorous analysis was done with the results achieved from those designs. Keeping overall objective of the study , prima-facie observation is , multiplicative inverse block is consuming most of the power . Any improvement in power signature should be made in that block only. RTL results give significant insight about different designs and as per the requirement of system, area , power , delay thing can be worked out . To sum up , implemented work provides minute detail about S-box and those can be used to improve the design.

REFERENCES

[1] Shanthini, M., Rajasekar, P., & Mangalam, H. "Design of low power S-box in Architecture Level using GF", *International journal of engineering research and general science (IJERG), 2014, pp-1–9*.

[2] Kumar, S., Sharma, V. K., & Mahapatra, K. K.."Low latency VLSI architecture of S-box for AES encryption", *International Conference on Circuits, Power and Computing Technologies (ICCPCT) 2013,*pp. 694-698.

[3] Oswald, E., Mangard, S., Pramstaller, N., & Rijmen, V. "A side-channel analysis resistant description of the AES S-box", *International Workshop on Fast Software Encryption, 2005*, pp. 413-423.

[4] Tiri, K., & Verbauwhede, I. "A VLSI design flow for secure side-channel attack resistant Ics", *Proceedings of the conference on Design, Automation and Test in Europe-Volume 3*, 2005, pp. 58-63.

[5] Zhou Y., & Feng, D."Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing", *IACR Cryptology ePrint Archive*, 2005, *pp* 388-389.

[6] Hossain FS, Ali ML "A Novel Byte-Substitution Architecture for the AES Cryptosystem", PLoS ONE Vol 10(10): e0138457,2015.

[7] Mui, Edwin NC, R. Custom, and D. Engineer. "Practical implementation of Rijndael S-box using Combinational logic." Custom R&D Engineer Texco Enterprise Pvt. Ltd, 2007.

[8] Zhang X, Parhi KK. "High-Speed VLSI Architectures for the AES Algorithm" IEEE Transaction on Very Large Scale Integration (VLSI) System, 2004, pp- 45–52.