IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834, p- ISSN: 2278-8735.
PP 73-79
www.iosrjournals.org

Reconfigurable Hardware Implementation Of Advanced Crypto System For Secured Communication

[1]Meenu Joy, [2]Aby Mathew
[1,2]Electronics and Communication Engineering Department,
Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala, India

***Abstract*:** *This paper proposes the implementation of high speed crypto system for secured communication that is hard to crack and improve the safety of data communicated. S-box that uses the combinational logic is introduced in this paper to improving the system computing speed instead of using ROM based LUTs. In this paper we synthesize both the S-boxes and compare the performance. Comparison shows that S-box using the combinational logic is faster. It is very efficient in terms of memory consumption. Using this method of encryption the data could be protected effectively..*

Keywords: *Add round key, Advanced Encryption Standard, Key expansion, Mix column, S-box, Shift row.*

I. INTRODUCTION

Wireless communication has become an inevitable thing in the present world. Confidentiality, access control, authentication and integrity are the important concern of the wireless communication.Federal department and other government agencies, electronic financial transactions, secure video surveillance systems etc. require protection of the data being transmitted through internet or any other kind of wireless medium from different type of malwares, third party attacks and various eavesdropping. Cryptography is such a kind of method that provides protection to the data communicated. It is the art of transforming confidential information into information which is incoherent to a third party. The confidential information is encrypted before transmission of the message and decrypted upon receipt using same secret keys.

In this paper we propose an advanced crypto system for the protection of data being communicated. It is a symmetric encryption system that uses the same private key for both encryption and decryption of data. The proposed crypto system replaces the existing S-box that is pre-computed values stored in a ROM based LUTs with an S-box that uses combinational logic [2]. We implement it on FPGA of family Spartan 6.

The rest parts of this paper are organized as follows: Section II describes the related works regarding the encryption method. In Section III, we presented an overview of encryption process. Section IV describes the new design method for S-box is proposed to solve the delay and boosting up the system. Section V describes the decryption process. VI presents results and comparison of both the method. Finally, we draw our conclusion in Section VII.

II. RELATED WORKS

Cryptography plays a significant role to maintain the integrity and confidentiality of the data being communicated. Many existing standard encryption algorithms and authentication schemes are available to efficiently defend against possible threats. The major classifications are symmetric cryptographies such as DES (Data Encryption Standard), Triple DES, AES (Advanced Encryption Standard), and asymmetric cryptography or public key encryption.

Asymmetric cryptography or public key encryption [3], use public key for encryption and private key for decryption. It is based on some complicated mathematics. Also the computer has to work very hard and large quantities of encrypted data make the system to be very slow.

Symmetric cryptographies such as DES (Data Encryption Standard), Triple DES, AES (Advanced Encryption Standard) use the same private key to encrypt and decrypt a message. The Data Encryption Standard [4], converts the plain text into cipher text by means of some permutation and substitution. Its key size is too small and so is easy to crack. Also short block size. It is a slower process and takes computer processor time. Triple DES advanced version of DES, the DES cipher algorithm is applied three times to each data block. This would increase the key size of triple DES by 168 bits. But it is slow, especially in software. It is susceptible to cryptanalysis and small block size. Advanced Encryption Standard [1] also uses the same private key to encrypt and decrypt a message. It is a combination of both substitution and permutation. This is fast in both software and

hardware as well. It is more secure and is less susceptible to cryptanalysis. It supports larger key sizes and block size makes it less open to attacks.

Another type of encryption method is the dynamic secret based encryption [5], which generates a shared symmetric secret key using transmission errors and other random factors. Then we monitor the error retransmission to select a group of frames and these frames are hashed into dynamic secret to encrypt the data. In this method unreliability is more when compared to other algorithms. Also the dynamic Key changing is very slow.

III. OVERVIEW OF ENCRYPTION

Figure 3.1 shows the overview of encryption. The encryption process start by applying the plain text of 128 bit that is to be communicated over the network and secret key of 128 bit. Key size used specifies number of repetitions of transformation rounds that convert plaintext into the cipher text. Table I shows the number of repetitions of transformation rounds. Each round consists of several processing steps.

TABLE I
Number of repetitions of transformation rounds

Key size	No. of rounds
128	10
192	12
256	14

The data block of 4 columns of 4 bytes which is called as state is applied as the input plain text. It undergoes 10 rounds of transformation and we get the cipher text at the transmitter side. Then we will send this cipher text to the receiver side and there it is decrypted back to the plain text using the same private key. Among the 10 rounds, 9 rounds in which state undergoes byte substitution, shift rows, mix columns, add round key.

Last round has no mix column. Also add round key is the only stage that uses the secret key. The key is expanded into array of 32-bit words. Among that 4 words form round key in each round.

A. SUBSTITUTE BYTES

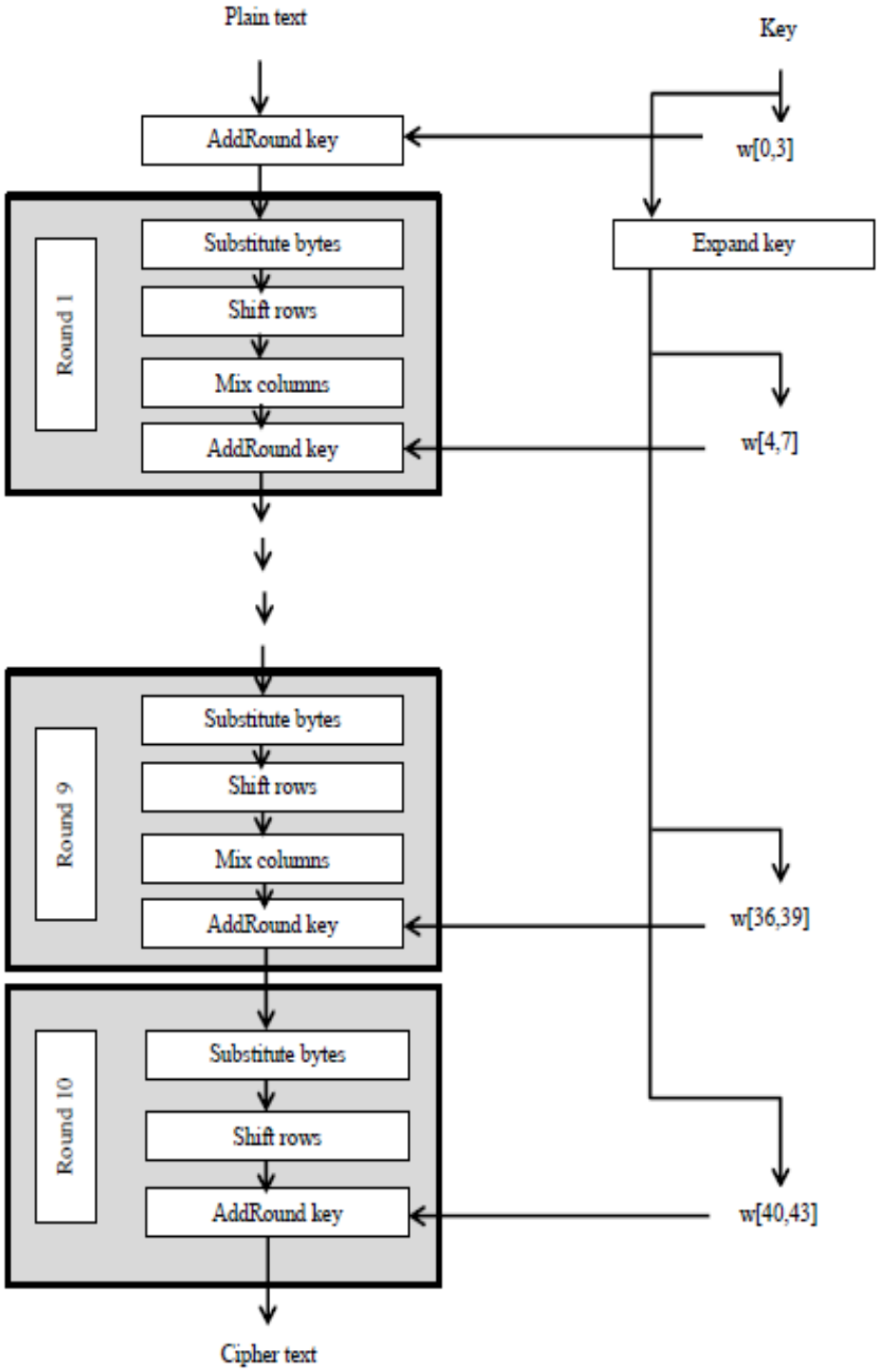
This stage of transformation is a table of 16x16 bytes containing a permutation of all 256 8-bit values. The table II shows the S-box. This S-box is a pre-computed values stored in a ROM based LUTs. It is designed to be resistant to all known attacks. Also it provides nonlinearity in cipher text.

B. SHIFT ROWS

Here we perform a circular byte shift in each row. The first row is kept unchanged. The second row is circular left shifted by 1 byte. The third row is circular left shifted by 2 byte. The fourth row is circular left shifted by 3 byte. This row shifting helps to avoid the columns being linearly independent. Also the AES is degenerated into four independent block ciphers. The decryption right shift operation.

C. MIX COLUMNS

In this stage each column is processed separately. Each byte replaced with value that is dependent on all 4 bytes in the entire column. Mix column provides diffusion in the cipher. And make it difficult to crack. This expresses each column as 4 equations too derive each new byte in column. Figure 3.2 shoes how to perform the mix column stage.



3.1 Overview of encryption process

TABLE II
S-box pre-computed values stored in ROM based LUTs

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

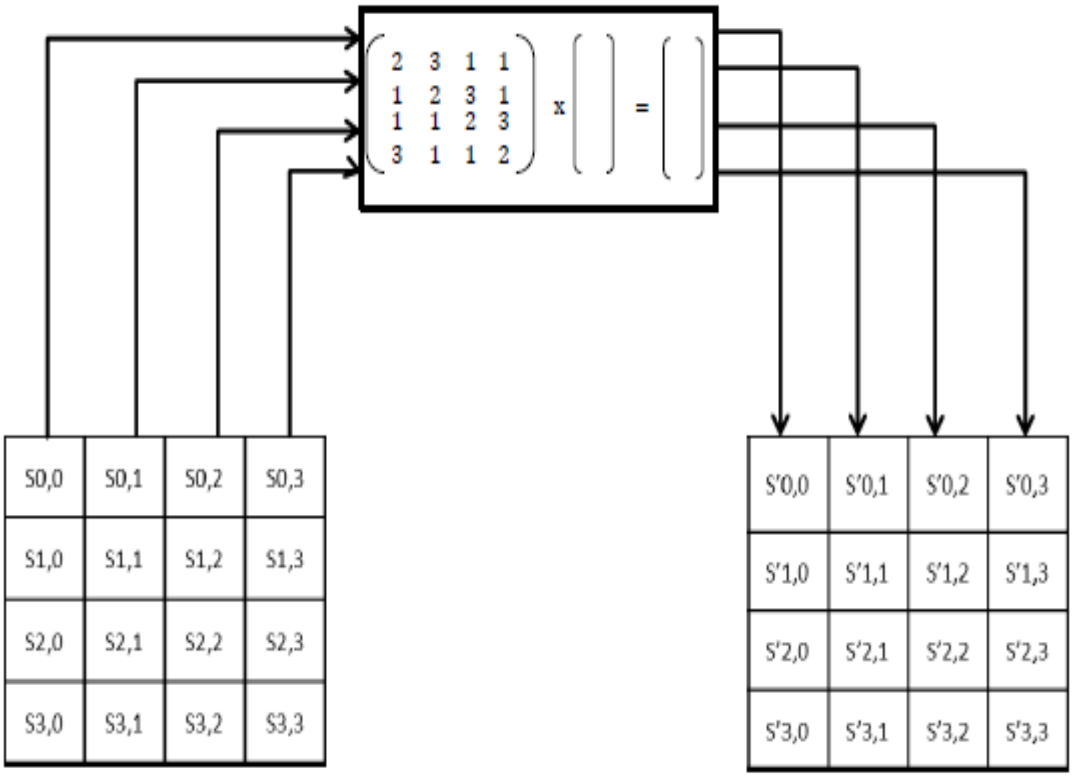


Fig 3.2 Mix column

D. ADD ROUND KEY

Here we perform XOR operation of state with 128-bits of the round key. It is designed to be as simple as possible. This is the only step uses the key. Using the key expansion the secret key is expanded and divided into

words, which words form round key in each round. Using this round key, add round key stage of transformation in each round is performed.

E. KEY EXPANSION

The secret key of 128 bit is expanded into array of 44 32-bit words. Among that 4 words form round key in each round. Using this round key, add round key stage of transformation in each round is performed.The key expansion starts by copying the key into first 4 words. Then creating groups of 4 words that depend on values in previous and four places back. The first word in 4 has to undergo rotate, S-box and XOR round constant on previous, before XOR fourth back. The figure 3.3 shows how the key expansion is performed.

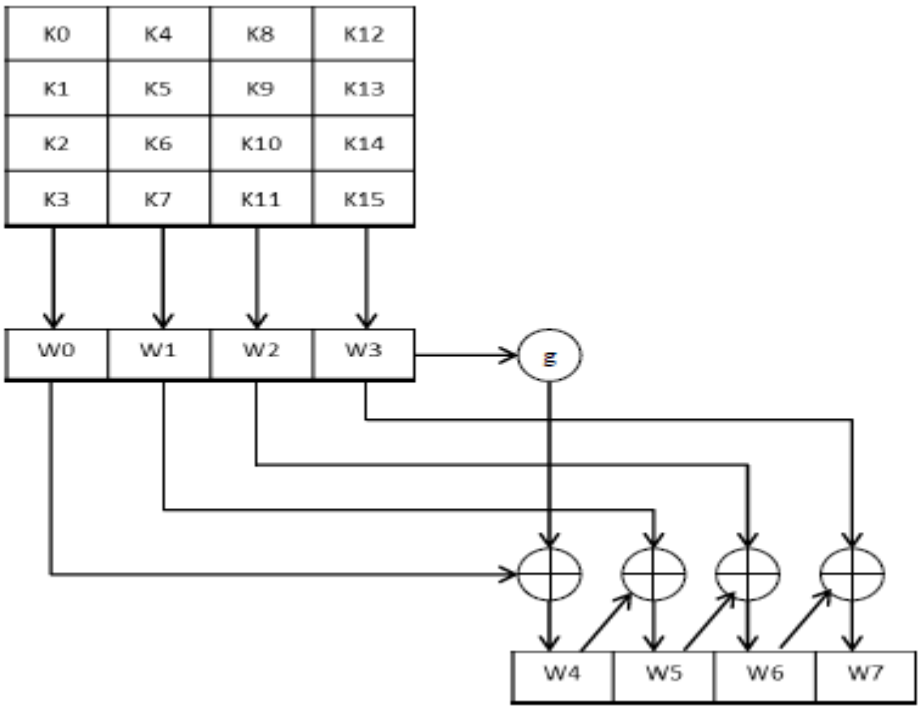


Fig 3.3 key expansion

IV. PROPOSED S-BOX

Existing S-box is a pre-computed values stored in a ROM based LUTs [1]. It has an unbreakable delay due to fixed access time for its read and write operation. Also it is expensive in terms of hardware. So we replace the ROM based LUTs with a combinational logic [2].

Here the SubByte is computed by taking the multiplicative inverse in GF(2^8) followed by an affine transformation. For its reverse, the InvSubByte transformation, the inverse affine transformation is applied first prior to computing the multiplicative inverse. The Affine Transformation and its inverse [2] can be represented in matrix form and it is shown below.

$$AT(a) = \begin{pmatrix} 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \\ 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \end{pmatrix} \times \begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad \text{---- (3.1)}$$

$$AT^{-1}(a) = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,1\,0\,0\,1 \\ 1\,0\,0\,1\,0\,1\,0\,0 \\ 0\,1\,0\,0\,1\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1 \\ 1\,0\,1\,0\,0\,1\,0\,0 \end{pmatrix} \times \begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{----- (3.2)}$$

Here vector ais the multiplicative inverse of the input byte from the state array. Byte representing a GF(2^8) element can be viewed as coefficients to each power term. For example, {10001011}2 is representing the polynomial q*7* + *q3* + *q* + *1* in GF(2^8).

Computation of the multiplicative inverse cannot be directly applied to an element. It has to be mapped to its composite field representation via an isomorphic function, δ. Likewise, after performing the multiplicative inversion, the result will also have to be mapped back from its composite field representation to its equivalent in GF(2^8) via the inverse isomorphic function δ^{-1}. Both δ and δ^{-1}can be represented as an 8x8 matrix [2].

Let q be the element in GF(2^8), then the isomorphic mappings and its inverse can be written as δ*q and δ^{-1}*q, which is a case of matrix multiplication as shown below, where q7 is the most significant bit and q0 is the least significant bit.

$$\delta \times q = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,1\,1\,1\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \end{pmatrix} \times \begin{pmatrix} q7 \\ q6 \\ q5 \\ q4 \\ q3 \\ q2 \\ q1 \\ q0 \end{pmatrix} \quad \text{----- (3.3)}$$

$$\delta^{-1} \times q = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0 \\ 0\,1\,1\,0\,0\,0\,1\,0 \\ 0\,1\,1\,1\,0\,1\,1\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 1\,0\,0\,1\,1\,1\,1\,0 \\ 0\,0\,1\,1\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,1\,0\,1 \end{pmatrix} \times \begin{pmatrix} q7 \\ q6 \\ q5 \\ q4 \\ q3 \\ q2 \\ q1 \\ q0 \end{pmatrix} \quad \text{----- (3.4)}$$

V. RESULT

Advanced crypto system is synthesized in Xilinx using Verilog programming language and implemented on FPGA of family Spartan 6. A 128 bit encryption and decryption is implemented and a comparison is made with S-box using ROM based LUTs and combinational logic.

Table III shows the performance comparison of crypto system with S-box using ROM based LUTs and combinational logic. From the table we can see that the number of slices LUTs have been reduced to half and the IO used is also reduced. And also we can see a significant reduction in delay. Delay reduced implies the speed of the system. While we are using the pre-computed values stored in a ROM based LUTs, there will be an unbreakable delay due to fixed access time for its read and write operation. The delay is reduced to around 58% and the memory utilization is also reduced. Since it is independent of clock, the process doesn't have to wait for the clock. It also improves the speed of the proposed system.

TABLE III
Performance comparison of crypto system with S-box using rom based LUTs and combinational logic.

Crypto system	**No. of Slices LUTs (Out of 9112)**	**No. of IO (Out of 232)**	**Delay (ns)**	**Memory Usage (KB)**
Using ROM	65	19	15.099	256036
Using Combinational Logic	33	16	8.781	204372

VI. CONCLUSION

In this paper high speed crypto system for secured communication is proposed using S-box that uses the combinational logic. We implemented the crypto system using both the S-box whose pre-computed values stored in a ROM based LUTs and combinational logic. Implementation results show that S-box that uses the combinational logic is more effective in terms of speed, memory and device utilization. Also it clock independent. Advanced crypto system can be used in application were speed and memory is given more importance.

REFERENCES

[1]. Vedkiran Saini, ParvinderBangar, Harjeet Singh Chauhan "Study and Literature Survey of Advanced Encryption Algorithm for Wireless Application", International Journal of Emerging Science and Engineering, ISSN: 2319–6378, Volume2, Issue-6, April 2014.
[2]. Edwin NC Mui, Custom R & D Engineer, Texco Enterprise Ptd. Ltd. "Practical Implementation of Rijndael S-Box Using Combinational Logic".
[3]. S. Nguyen and C. Rong, "ZigBee security using identity-based cryptography autonomic and trusted computing," in *Proc. 4th Int. Conf. Autonomic Trusted Comput. (ATC'07)*, 2007, vol. 4610, Lecture Notes in Computer Science, pp. 3–12.
[4]. Davis, R. Secretary of Defence for Research and Engineering "The data encryption standard in perspective", Communications Society Magazine, IEEE (Volume:16 Issue: 6), 06 January 2003.
[5]. Ting liu, yangliu, yashanmao, yao sun, xiaohongguan, Weibo gong and sheng xiao "A dynamic secret-based encryption scheme for Smart grid wireless communication", IEEE transactions on smart grid, vol. 5, no. 3, may 2014.
[6]. Husheng li, shuping gong, lifenglai, zhuhan, robert c. Qiu and depeng yang "Efficient and secure wireless communications for Advanced metering infrastructure in smart grids", IEEE transactions on smart grid, vol. 3, no. 3, september 2012.
[7]. Xudongwangand ping yi "Security framework for wireless communications in Smart distribution grid", IEEE transactions on smart grid, vol. 2, no. 4, december 2011.
[8]. Jinyuexia and yonggewang "Secure key distribution for the smart grid", IEEE transactions on smart grid, vol. 3, no. 3, september 2012
[9]. Guest editorial Cyber, physical, and system security For smart grid, IEEE transactions on smart grid, vol. 2, no. 4, december 2011.
[10]. Dapengwu *and* chi zhou "Fault-tolerant and scalable key Management for smart grid", IEEE transactions on smart grid, vol. 2, no. 2, june 2011.
[11]. Ye yan, yiqian and hamidsharif "A secure data aggregation and dispatch scheme for home area networks in smart grid", 2011 IEEE transactions on smart grid, vol. 1, no. 1, September 2010
[12]. Anthony r. Metke and randy l. Ekl "Security technology for smart grid networks", IEEE transactions on smart grid, vol. 1, no. 1, june 2010.
[13]. Fengjun li, boluo and pengliu "Secure information aggregation for smart grids Using homomorphic encryption", IEEE transactions on smart grid, vol. 1, no. 1, june 2009.