BẢO MẬT ẢNH DỰA TRÊN HỘP THAY THẾ VÀ LÝ THUYẾT HỖN ĐỘN

IMAGE ENCRYPTION BASED S-BOX AND MULTI CHAOS

Nguyễn Anh Dũng[1,*], Ngô Văn Huấn

TÓM TẮT

Bài báo đề xuất thiết kế hộp thay thế S-box mới sử dụng Triangle-Chaotic (TCM), Logistic 1D cải tiến. Đề xuất phương án bảo mật ảnh dùng hàm Chaos 2D cross để xáo trộn vị trí điểm ảnh, thay đổi giá trị điểm ảnh sử dụng S-box và Logistic 1D. Sử dụng các tiêu chuẩn đánh giá chất lượng ảnh bảo mật để kiểm tra phương án đề xuất, kết quả chỉ ra phương án đề xuất đảm bảo yêu cầu.

Từ khóa: *Chaos, S-box, Logistic, Triangle-Chaotic(TCM), bảo mật ảnh, hộp thay thế.*

ABSTRACT

In this paper, a new S-box is proposed base on combination Triangular Chaotic Map (TCM) and improved 1D Logistic Chaos. Then the shuffled image is used by 2D cross Chaos, diffusion of image pixel using S-box and Logistic 1D. After analysis and test, the algorithm satisfies the requirments of safety image encryption.

Keywords: *image encryption, S-box, 2D cross map, Triangular Chaotic Map (TCM), 2D Cat, Improved 1D Logistic.*

[1]Khoa Điện tử, Trường Đại học Công nghiệp Hà Nội
[2]Khoa Điện tử, Học viện Kỹ thuật Quân sự
*Email: anhdung0412@gmail.com
Ngày nhận bài: 28/12/2017
Ngày nhận bài sửa sau phản biện: 27/3/2018
Ngày chấp nhận đăng: 21/8/2018
Phản biện khoa học: TS. Hà Mạnh Đào

1. GIỚI THIỆU

Trong những năm gần đây lý thuyết hỗn độn (Chaos) và hộp thay thế (S-box) được đề xuất và sử dụng nhiều trong bảo mật nói chung và bảo mật ảnh nói riêng. Đặc tính của Chaos là: tạo ra các chuỗi số ngẫu nhiên, nhạy cảm với các tham số đầu vào như giá trị khởi tạo và các tham số điều khiển. Nhiều hệ thống bảo mật ảnh dựa trên các hàm Chaos khác nhau, hàm Logistic [1-3], Barker [4], Arnold [14], Ikeda [8]; việc kết hợp S-box và Chaos trong bảo mật ảnh cũng đã được nghiên cứu. Thiết kế S-box cũng được nghiên cứu nhiều: Chen đề xuất phương án thiết kế S-box sử dụng hàm Baker 3 chiều [4]; Muhammad Asim sử dụng hàm PLCM để tạo hàm S-box[5]; năm 2013 Mona Dara và Koorous Manochehri đề xuất phương án thiết kế S-box dựa trên khóa bảo mật và hàm Logistic [11]; Cristian-Iulian Rîncu, Vasile-Gabriel IANA 2014 kết hợp nhiều hệ thống Chaos là Logistic, Tent, PLCW khác nhau để tạo ra S-box [9]; F. J. Luma1 đề xuất phương án tạo S-box sử dụng một số hàm Chaos 2D cross, 2D logistic và 2D Cat để tạo S-box [14].

Bài báo đề xuất phương án sử dụng hàm Triangle-Chaotic Map(TCM) [12] và Logistic cải tiến [13] để tạo S-box. Sử dụng hàm Chaos 2D cross để thay đổi vị trí điểm ảnh, thay đổi giá trị điểm ảnh sử dụng S-box và Chaos Logistic 1D. Sử dụng các tiêu chuẩn đánh giá chất lượng ảnh bảo mật để kiểm tra phương án đề xuất.

2. CƠ SỞ LÝ THUYẾT

2.1. 2D Cat

Hàm 2D Cat sử dụng theo công thức sau:

$$\begin{cases} x_{i+1}=(x_i+a*y_{i+1}) \\ y_{i+1}=(b*x_i+(a*b+1)*y_i) \end{cases} \quad (1)$$

Trong đó: a và b là hai tham số điều khiển, (x_i,y_i) và (x_{i+1},y_{i+1}) là giá trị thứ i và i+1. Chaos này được sử dụng để xáo trộn vị trí điểm ảnh theo công thức (2), với (x_i,y_i) và (x_{i+1},y_{i+1}) là vị trí điểm ảnh thứ i và i+1, N là độ rộng ảnh.

$$\begin{cases} x_{i+1} = (x_i + a * y_{i+1}) \bmod N \\ y_{i+1} = (b * x_i + (a * b + 1) * y_i) \bmod N \end{cases} \quad (2)$$

2.2. Logistic map

Hàm một chiều logistic được biểu diễn như sau:

$$x_{n+1} = \alpha * x_n * (1 - x_n) \quad (3)$$

Trong đó: x_n là giá trị thứ n, $x_n \in [0; 1]$; α là tham số điều khiển giá trị $\alpha \in (0; 4]$, khi $\alpha \in (3,6; 4]$, thì hàm Logistic trở thành hàm Chaos.

2.3. Logistic map cải tiến [13]

Hàm Logistic cải tiến được biểu diễn như sau:

$$x_{n+1} = L(\alpha, x_n) * G(m) - \text{floor}(L(\alpha, x_n) * G(m)) \quad (4)$$

Trong đó:

$$\begin{cases} L(\alpha, x_n) = \alpha * x_n * (1 - x_n) \\ G(m) = 2^m \quad m \in Z^+ , \ m \geq 8 \end{cases} \quad (5)$$

G(m) là hàm điều chỉnh với tham số điều chỉnh *m*. Giá trị *m* càng lớn đặc tính chaos càng được thể hiện rõ. Hàm floor được sử dụng để đảm bảo giá trị của $x_n \in [0; 1]$.

2.4. Triangle-Chaotic Map(TCM) [12]

Hàm Triangle-Chaotic Map [12] biểu diễn như sau:

$$x_{n+1} = \begin{cases} (t * x_n * (1 - x_n)) \bmod 1 & n \bmod 2 = 0 \\ (\pi * \beta * x_n) \bmod 1 & n \bmod 2 = 1 \end{cases} \quad (6)$$

Trong đó: x_n là giá trị thứ n, $x_n \in [0; 1]$, t và β là tham số điều khiển giá trị $t \in (0; +\infty]$, β là số lẻ và $\beta \in [3; 99]$.

2.5. 2D cross

Hàm 2D cross được biểu diễn như sau:

$$\begin{cases} x_{i+1} = 1 - \mu y^2 \\ y_{n+1} = \cos(k.\cos^{-1} x_n) \end{cases} \quad x, y \in [-1, 1] \quad (7)$$

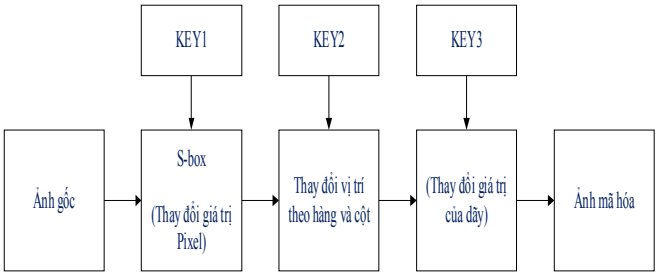
Trong đó: μ và k là 2 tham số điều khiển độc lập, x_n và x_{n+1} là giá trị thứ i và i+1. Khi μ = 2 và k = 6 thì hệ thống trở thành hệ thống Chaos động có tính bảo mật cao [8].

3. ĐỀ XUẤT MÔ HÌNH MÃ HÓA VÀ GIẢI MÃ ẢNH

3.1. Mô hình mã hóa: Sơ đồ mã hóa ảnh đề xuất trên hình 1, quá trình mã hóa (bảo mật ảnh) được chia làm 3 bước với chức năng khác nhau, sử dụng ba KEY khác nhau để mã hóa cho từng bước.

Bước 1: Thiết kế S-box sử dụng KEY1, 2 hàm Chaos là TCM và Logistic cải tiến được tiến hành theo các bước như sau:

Bước 1.1: Thiết lập các giá trị khởi tạo cho các hàm ngẫu nhiên TCM (6) và Lo (4). Tiến hành chạy các hàm TCM và Logistic cải tiến N giá trị trong đó N là hằng số. Thu được 2 chuỗi số TCM(1…N) và IM(1…N).



Hình 1. Sơ đồ mã hóa ảnh

Bước 1.2: Lấy 2 giá trị TCM(N) và IM(N) làm giá ban đầu để khởi tạo hàm TCM và Logistic cải tiến một lần ta thu được TCM(1) và IM(1).

Bước 1.3: Chuyển giá trị của TCM(1) và IM(1) sang dạng số nguyên không dấu 8 bit. Theo công thức sau:

$$\begin{cases} num1 = \bmod(TCM(1) * 10^{16}), 256) \\ num2 = \bmod(IM(1) * 10^{16}), 256) \end{cases} \quad (8)$$

Bước 1.4: Thực hiện XOR 2 giá trị

Num = num1 XOR num2

Bước 1.5: Kiểm tra giá trị Num với các phần tử đã được lưu trong S-box. Nếu Num không trùng với giá trị nào nằm trong S-box thì Num sẽ được thêm vào S-box. Nếu trùng với giá trị nào thì Num sẽ bị loại bỏ.

Bước 1.6: Quay trở lại thực hiện từ bước số 1.2 cho đến khi S-box đầy đủ các giá trị từ 0 đến 255.

Bước 1.7: Quá trình thay thế giá trị của các điểm ảnh sử dụng S-box như sau:

Giá trị của từng điểm ảnh đầu vào (8 bit) tách thành hàng (H) gồm 4 bít cao và cột C gồm 4 bít thấp.

Thực hiện tìm hàng và cột mới theo công thức (2) trong đó H tương ứng với x và C tương ứng với y

Dựa vào giá trị hàng và cột mới tra bảng S-box ta tìm được giá trị thay thế.

Ma trận của ảnh gốc đầu ra S-box là $P_{M \times N}$ với M số hàng, N số cột

Bước 2: Sử dụng hàm 2D Cross với khóa KEY2 để tiến *hành xáo trộn vị trí của hàng và xáo trộn vị trí cột* theo thuật toán sau:

Xáo trộn vị trí của hàng theo các bước sau:

Bước 2.1: KEY2 có giá trị khởi tạo x_0, y_0, μ = 2 và k = 6. Tiến hành lặp K lần theo công thức (7) (với giá trị K = (100 XOR S-box(16,16)) ta thu được dãy tam X0 = {x_1, x_2,…x_K}, tam Y0 = {y_1, y_2,…y_K}. Tiếp tục lặp theo công thức (7) với giá trị khởi tạo mới là $x_0 = x_K$, $y_0 = y_K$ thu được dãy X0 = {x_1, x_2,…x_N}, Y0 = {y_1, y_2,…y_N} (với N là số cột của ảnh, $x_i \neq x_j$ và $y_i \neq y_j$ với $\forall$ i, j)

Bước 2.2: Sắp xếp dãy X0, Y0 theo thứ tự tăng dần (tính hạng của các x_i và y_i) ta được 2 dãy mới tương ứng M_X0 = {z_1, z_2,…z_N} và M_Y0 = {g_1, g_2,…g_N}. Tạo dãy Mx và My lưu hạng của x_i và y_i.

Bước 2.3: Xáo trộn hàng 1 theo dãy Mx, xáo trộn hàng 2 theo dãy My. Tiếp tục lặp từ bước 2.1 đến bước 2.3 đến hết các hàng.

Xáo trộn vị trí của cột theo cácbước sau: Tương tự như các bước của hàng.

Kết quả đầu ra bước 2 ta có ma trận $G_{M \times N}$

Bước 3: Thực hiện thay đổi giá trị các điểm ảnh dựa vào hàm Logistic theo thuật toán như sau:

Bước 3.1: Từ giá trị KEY3 ta có giá trị khởi tạo α và x_0, tiến hành lặp K lần theo công thức (3) (với giá trị K = (100 XOR $G_{M \times N}$ (M,N)) ta thu được dãy X0 = {x_1, x_2,…x_K}

Bước 3.2: Lấy giá trị x_K ở bước 2 làm giá trị khởi tạo x_0. Thực hiện hàm (3) N lần ta thu được chuỗi ngẫu nhiên P(p_1…p_N).

Bước 3.3: Tại hàng 1: các vị trí cũng tương ứng là (p_1…p_N), với từng giá trị p_i các phép toán mã hóa sau:

Giá trị của P_i	Phép mã hóa tương ứng
$P_i \in (0; 0,3]$	MOD((p_i*10^{15}), 256) XOR $G_{M \times N}$ (i,j)
$P_i \in (0,3; 0,6]$	NOT(MOD((p_i*10^{15}), 256)) XOR $G_{M \times N}$ (i,j)
$P_i \in (0,6; 1)$	MOD((p_i*10^{15}), 256) XOR NOT($G_{M \times N}$ (i,j))

Lặp lại bước 3.2 với giá trị khởi tạo $x_0 = p_N$ đến hết hàng cuối cùng.

3.2. Quá trình giải mã: các bước ngược với quá trình mã hóa ảnh.

4. KẾT QUẢ VÀ ĐÁNH GIÁ

Kết quả ảnh mã hóa và giải mã:

Việc thực hiện thuật toán đề xuất được tiến hành trên ảnh lena.jpg, BVB.jpg, chelsea.jpg., khóa mã hóa cho các KEY như sau:

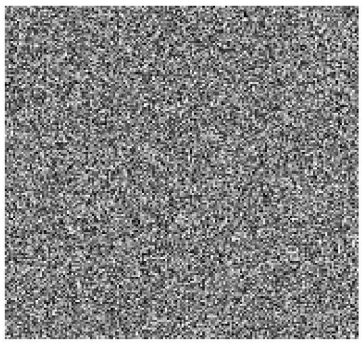
KEY1: $x_0 = 0,345$; $t = 4$; $\beta = 13$; $m = 9$, $\alpha_1 = 3,7890123456$;

KEY2: $x_0 = 0,345$, $y_0 = -0,5678901234$;

KEY3: $x_0 = 0,345$; $\alpha_2 = 3,9012345678$;

Hình 2 ảnh lena trước, sau mã hóa và giải mã đúng theo các khóa ở trên

Ảnh trước mã hóa



Ảnh sau mã hóa

Ảnh giải mã đúng

Hình 2. Ảnh lena.jpg trước, sau mã hóa giải mã đúng

Phân tích độ nhạy của khóa: phân tích độ nhạy của khóa S-box và độ nhạy ảnh giải mã với các khóa.

***- Phân tích độ nhạy của khóa S-box*:** Thuật toán sử dụng bộ khóa KEY1 để tạo Sbox. KEY1: x_0, t, β, m và α_1. Mô phỏng sử dụng tham số sau: KEY1: $x_0 = 0,345$; $t = 4$; $\beta = 13$; $m = 9$, $\alpha_1 = 3,7890123456$; ta thu được S-box có giá trị như sau:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	22	181	212	66	233	231	177	129	4	73	146	226	60	19	58	51
1	87	203	45	196	161	63	154	33	112	143	188	239	1	49	43	3
2	206	98	80	230	149	107	57	244	132	71	166	37	172	113	201	15
3	64	85	2	119	225	134	254	52	116	27	40	167	187	101	220	6
4	78	69	65	251	76	171	35	163	5	23	127	74	208	55	61	75
5	90	126	238	179	20	253	217	142	189	67	110	81	249	250	47	133
6	153	190	136	94	56	109	79	95	50	31	92	29	227	224	198	182
7	168	68	205	221	141	240	237	86	123	46	12	202	178	62	77	194
8	248	121	193	25	151	93	138	162	124	185	8	213	16	96	247	176
9	100	18	26	104	17	155	245	169	128	246	91	145	255	147	235	82
10	209	130	236	184	243	223	38	170	59	111	21	9	140	186	152	0
11	70	28	174	84	48	42	34	32	41	173	120	13	252	242	218	158
12	24	180	232	216	10	54	192	115	7	210	200	219	175	160	30	97
13	125	228	183	215	199	103	211	195	99	53	108	102	114	157	89	150
14	131	207	106	72	36	241	156	144	122	88	117	234	191	165	148	39
15	44	164	11	118	214	204	197	139	137	135	222	14	105	83	159	229

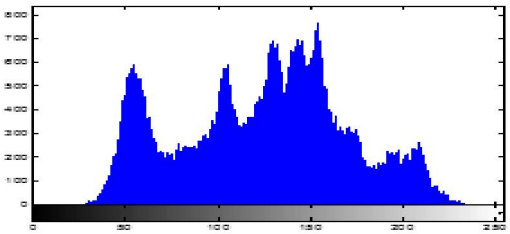
Khi thay đổi lần lượt các giá trị trong khóa KEY1 với giá trị KEY1 khác nhau. Giá trị khởi tạo x_0 mới $x_0 = 0,34500000000000\underline{\mathbf{1}}$ hoặc $t = 4,00000000000\underline{\mathbf{1}}$ giá trị $\beta = 13$ sang $\beta = 15$ giá trị $m = 9$ sang $m = 10$, $\alpha_1 = 3,789012345\underline{\mathbf{6}}$ thành $\alpha_1 = 3,789012345\underline{\mathbf{5}}$ ta đều thu được các Sbox khác nhau dẫn đến giải mã đều không chính xác.

- Độ nhạy ảnh giải mã với các khóa:

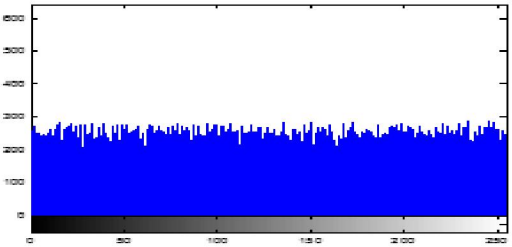
KEY2 thay đổi giá trị $y_0 = -0,567890123\underline{\mathbf{4}}$ thành $y_0 = -0,567890123\underline{\mathbf{5}}$, KEY3 thay đổi $\alpha_2 = 3,901234567\underline{\mathbf{8}}$ thành $\alpha_2 = 3,901234567\underline{\mathbf{9}}$ thì kết quả giải mã sai.

Phân tích Histogram

Historgram phản ánh phân bố thống kê giá trị của các điểm ảnh, ảnh bảo mật tốt histogram có phân bố đều. Histogram của ảnh trước và sau mã hóa trên hình 3; Ảnh sau mã hóa phân bố tương đối đồng đều.



Histogram ảnh lena.jpg



Histogram ảnh lena.jpg sau mã hóa

Hình 3. Histogram ảnh trước và sau mã hóa

Phân tích tương quan [13]

Các điểm ảnh lân cận nhau của ảnh gốc thường có hệ số tương quan cao. Ảnh sau mã hóa yêu cầu hệ số tương quan giảm tối thiểu. Các tham số hệ số tương quan theo chiều ngang, chiều dọc và chéo được dùng để đánh giá chất lượng của ảnh sau bảo mật. Trong bài báo sử dụng 5000 điểm ảnh dùng để tính tương quan theo công thức (9).

Bảng 1 chỉ ra hệ số tương quan giữa ảnh gốc và ảnh mã hóa.

$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i \;;\quad D(x)=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E(x)\right)^2$$
$$\operatorname{cov}(x,y)=\frac{1}{P}\sum_{i=1}^{N}\left(x_i-E(x)\right)\left(y_i-E(y)\right);\quad r_{xy}=\frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \qquad (9)$$

Bảng 1. Bảng hệ số tương quan giữa ảnh gốc và ảnh mã hóa

Ảnh	Trước mã hóa			Sau khi mã hóa		
	Chiều ngang	Chiều dọc	Đường chéo	Chiều ngang	Chiều dọc	Đường chéo
lena.jpg	0,91385	0,95154	0,88882	0,008512766	-0,0017926	0,003373747
BVB.jpg	0,95353	0,964006	0,93004	-0,00490436	-0,0167341	0,001614163
Chelsea.jpg	0,95275	0,95651	0,92247	-0,0122402	-0,0158240	-0,00656629

NPCR đánh giá sự thay đổi giữa các pixel [10]

NPCR được sử dụng để đánh giá sự thay đổi giữa ảnh gốc và ảnh mã hóa. Ta có ảnh gốc và ảnh mã hóa là C1 và C2 tương ứng. Định nghĩa mảng 2 chiều D có kích thước tương ứng với C1 và C2. Giá trị D(i,j) được xác định từ các giá trị của C1(i,j) và C2(i,j) tương ứng theo như sau:

Nếu C1(i,j) = C2(i,j) khi đó D(i,j) = 1; C1(i,j) ≠ C2(i,j) khi đó D(i,j) = 0;

Tham số NPCR được định nghĩa như sau:

$$NPCR=\frac{\sum_{ij}D(i,j)}{M*N} \qquad (10)$$

Trong đó M và N là kích thước của ảnh. Giá trị của NPCR càng lớn độ bảo mật ảnh càng cao trường hợp lý tưởng NPCR = 1.

Bảng 2. Các giá trị NPCR với một số ảnh mẫu

Ảnh	Giá trị NPCR
Lena.jpg	0,996398925781250
BVB.jpg	0,995614814814815
Chelsea.jpg	0,996133609693878

5. KẾT LUẬN

Qua các kết quả đánh giá ở trên phương án bảo mật ảnh đề xuất đảm bảo yêu cầu của ảnh bảo mật. Không gian khóa ảnh bảo mật lớn, đặc biệt có không gian khóa KEY1 dùng tạo S-box sử dụng Triangle-Chaotic Map và Logistic cải tiến rất lớn.

TÀI LIỆU THAM KHẢO

[1]. Jakimoski G, Kocarev L. *Chaos and cryptography*: block encryption ciphers. IEEE Trans Circ Syst-I 2001;48(2):163–9 boxes based on chaotic maps, Chaos Solitons Fractals, 23(2): 413-419.

[2]. Tang, G., X. Liao and Y. Chen, 2005. *A novel method cryptography for designing S-boxes based on chaotic maps*, Chaos 40(1): 505-519. Solitons Fractals, 23(2): 413-419.

[3]. Tang, G. and X. Liao, 2005. *A method for dynamical S-boxes based on discretized chaotic map*, Chaos Solitons Fractals, 23(5): 1901-1909.

[4]. Chen, G., Y. Chen and X. Liao, 2007. *An extended method for obtaining S-boxes based on three- dimensional chaotic baker maps*, Chaos Solitons Fractals, 31(3): 571-579.

[5]. Muhammad Asim et al. *Efficient and Simple Method for Designing Chaotic S-Boxes*, ETRI Journal, Volume 30, Number 1, February 2008.

[6]. Ruming Yin, Jian Yuan, Jian Wang, Xiuming Shan, Xiqin Wang, *Designing key-dependent chaotic S-box with larger key space*, Chaos, Solitons and Fractals 42 (2009) 2582–2589.

[7]. J. Peng, S. Jin, L. Lei, R. Jia, *A Novel Method for Designing Dynamical Key-Dependent S-Boxes based on Hyperchaotic System*, International Journal of Advancements in Computing Technology(IJACT) .

[8]. Xiaogang Jia, *Image Encryption using the Ikeda map*, 2010 International Conference on Intelligent Computing and Cognitive informatics

[9]. D.Chattopadhyay, M.K. Mandal and D.Nandi, *Symmetric key chaotic image encryption using circle map*, Indian Journal of Science and Technology Vol.4 No.5 (May 2011)

[10]. Xiaoling Huang, *A New digital image encryption algorithm based on 4D chaotic system*, International Journal of Pure and Applied Mathematics Vol. 80 No. 4 2012, 609-616

[11]. Mona Dara and Kooroush Manochehri, *A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key*, World Applied Sciences Journal 28 (12): 2003-2009, 2013

[12]. Mahmoud Maqableh, *A Novel Triangular Chaotic Map (TCM) with Full Intensive Chaotic Population Based on Logistic Map*, Journal of Software Engineering and Applications, 2015, 8, 635-659

[13]. Liu Rui, *New Algorithm for Color Image Encryption Using Improved 1D Logistic Chaotic Map*, The Open Cybernetics & Systemics Journal, 2015, Volume 9 211

[14]. F. J. Luma1, H. S. Hilal and A. Ekhlas, *New Dynamical Key Dependent S-Box based on chaotic maps,* IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. IV (July - Aug. 2015), PP 91-101