Data Encryption with Linear Feedback Shift Register

Subhra Mazumdar , Tannishtha Som

Abstract— A data encryption technology which ensures secrecy of the data while being transferred over a long distance. It can provide about 80-85% data security as decoding of data involves inverting the feedback function or generating the binary sequence which will help in retrieving the data after some recombination operation.

Index Terms— octal word time generation , linear feedback shift register, feedback function, data security, priority encoder, email server, SMTP(simple mail transfer protocol) ,POP(post office protocol) , device sensitive password check.

1 INTRODUCTION

An efficient method to modify the plain text into an encoded cipher text , not easily predictable ensuring that the key value is irrecoverable when data is attacked while being transmitted. If a data is lost or extra bit gets added while transmission, the system will automatically show error as all the processes are synchronised. To avoid data being modified while transmission, different types of feedback function for 100 characters(3-bit sequence specific and different for adjacent row and column input devices in the register shown in figure 4; arranged in a 10 * 10 matrix) having different bit sequence is devised. Two stage password check(one of them being device specific) is used for decoding the message.

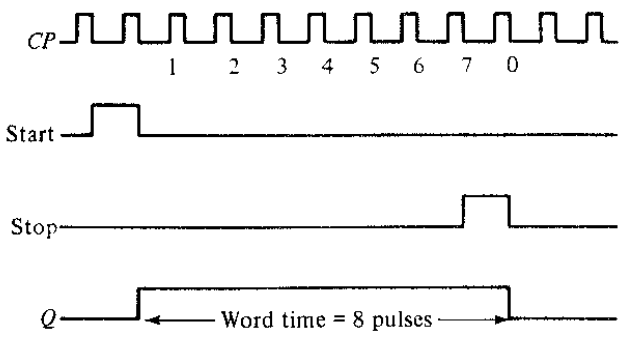

figure 1: OCTAL WORD-TIME SIGNAL GENERATION

2 PURPOSE AND DESIGN OF THE DEVICE

Converting the data to its ASCII value, one character at a time, using a 2^8 x 8 priority encoder (1 byte per character), the 8-bit sequence is stored in an 8-bit right shift register M (PARALLEL IN).

Then a shift control input is introduced with the clock pulse having an octal word-time signal so that number pulse is equal to the number of bits in the shift register. (word - time signal-figure 1, circuit diagram-figure 2).

The shift register has got a 0-bit feedback(in figure 3). When the input bits are shifted towards right, 0-bit enters from the leftmost register so that at the end of the 8th clock pulse the content of 8-bit register is refreshed back to 0. The output mode is SERIALLY OUT.

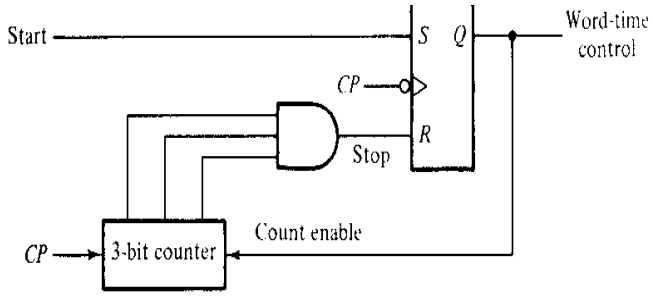

figure 2: CIRCUIT DIAGRAM FOR SHIFT CONTROL

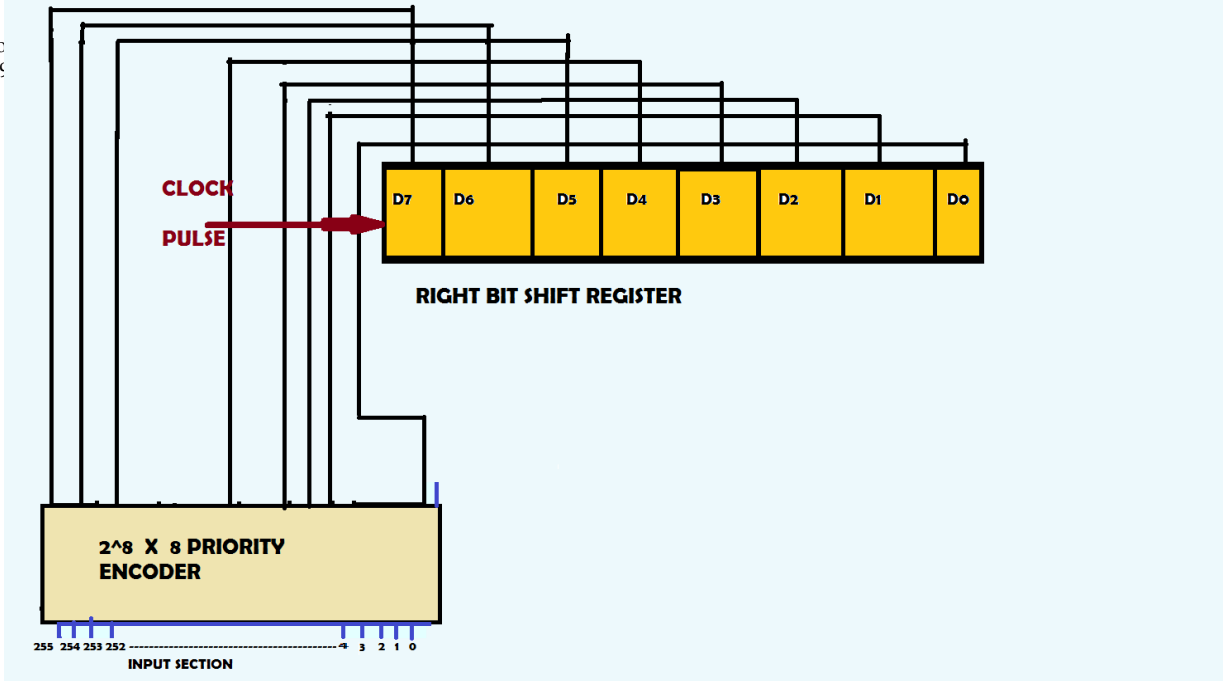

figure 3- REGISTER 'M'

3 FEEDBACK FUNCTION

After the first clock pulse, the extreme right bit of the register is made to undergo some transformation caused by the linear feedback shift register (LFSR) set-up.

In reference to figure 4, a 3-bit register having an initial content set to 000 is used .Let the 3-bit register (SERIALLY IN SERIALLY OUT) from left to right be named as A, B and C. The 8-bit right shift register (figure 3) and the 3-bit LFSR are working under the same clock pulse or timing sequence. The initial output of C is XOR-ed with the first bit shifted out of the 8-bit shift registers.

The feedback function which supplies input to A is f =((AB XNOR C) XOR A).(ABC)'. This is continued for 8-clock pulses where we obtain the output from C as given by table 1:-

C0	C1	C2	C3	C4	C5	C6	C7
0	0	0	1	0	1	1	1

Simultaneously output from the 8-bit shift register is XOR-ed with the output from C.

For example, suppose you want to transfer the character 'b' whose ASCII value is 98.When decoded gives the bit sequence 01100010.

01100010 is transferred to the 8-bit right shift register and is bits are serially transferred one by one. The shifting of bits is shown in table 2.

Table 3 shows how at each clock pulse C is XOR-ed with X. Let it be Z.

Figure 4 shows the internal mechanism providing the necessary data security by modifying the original data.

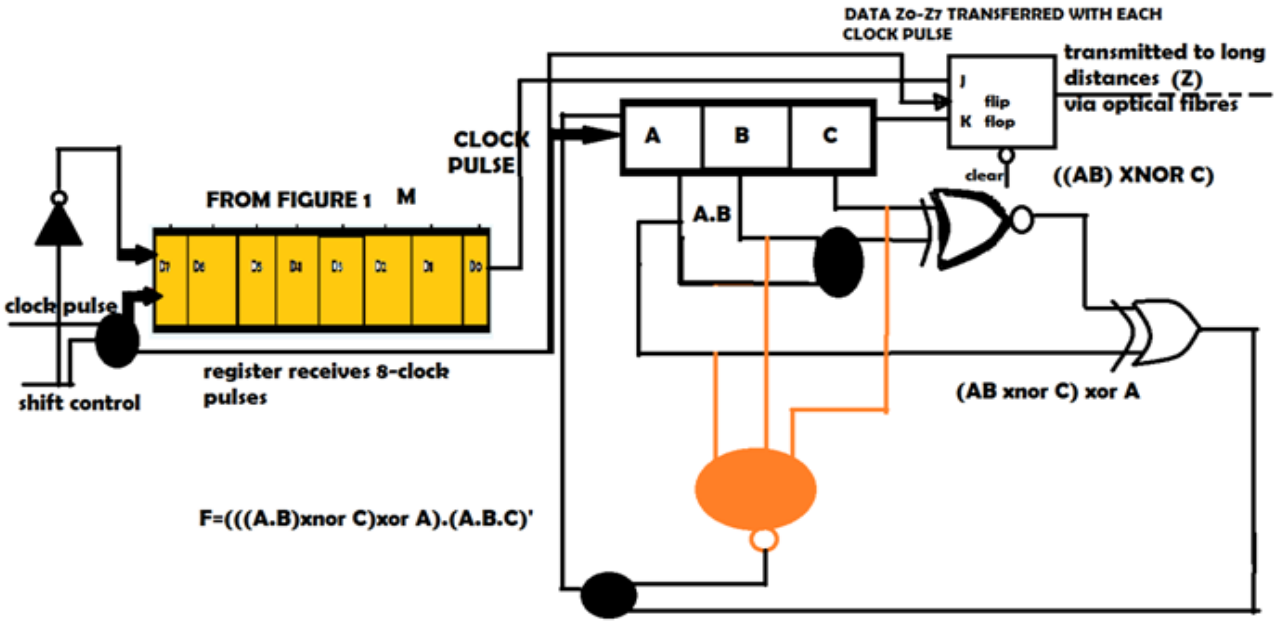

figure 4: CONVERTING ORIGINAL BIT SEQUENCE TO CIPHER BITS

At clock pulse:

1	2	3	4	5	6	7	8
C0	C1	C2	C3	C4	C5	C6	C7
0	0	0	1	0	1	1	1
X0	X1	X2	X3	X4	X5	X6	X7
0	1	0	0	0	1	1	0
Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7
0	1	0	1	0	0	0	1

4 TABLES

TABLE 1:- FOR THE SEQUENCE GENERATED IN REGISTER A, B, C BY FEEDBACK FUNCTION

CLOCK PULSE	Sequence A B C	f=((AB XNOR C) XOR A).(ABC)'
Initially at t=0	C0 0 0 0	1
After t=1	C1 1 0 0	0
After t=2	C2 0 1 0	1
After t=3	C3 1 0 1	1
After t=4	C4 1 1 0	1
After t=5	C5 1 1 1	0
After t=6	C6 0 1 1	0
After t=7	C7 0 0 1	0
After t=8	0 0 0	1

TABLE 2:- FOR SHIFTING OF DIGITS IN THE 8-BIT SHIFT REGISTER FOR CHARACTER 'B' – ASCII VALUE-98-BINARY –(01100010)

CLOCK PULSE	BIT SEQUENCE	OUTPUT TO BE XOR-ED WITH C
Initially at t=0,	Contents inserted	Bits shifted out(X)
At 1	0 1 1 0 0 0 1 0	0
At 2	0 0 1 1 0 0 0 1	1
At 3	0 0 0 1 1 0 0 0	0
At 4	0 0 0 0 1 1 0 0	0
At 5	0 0 0 0 0 1 1 0	0
At 6	0 0 0 0 0 0 1 1	1
At 7	0 0 0 0 0 0 0 1	1
At 8	0 0 0 0 0 0 0 0	0

The encoded data goes to user's system and saved in any server(say a MICROSOFT SERVER) so that the receiver can view the message from anywhere where there is a computer and internet connection . User can view the sender's name in any computer but the text part of the mail is shown in the sequence of 0's and 1's (i.e. , all the unique sequence of Z0-Z7 is visible for individual character) .

TABLE 3:- BIT SEQUENCE TRANSFERRED ONE AT A TIME

CLOCK PULSE	Output from C(C0-C7)	Bit output –X (X0-X7)	Z=C XOR X (Z0-Z7)
1	0	0	0
2	0	1	1
3	0	0	0
4	1	0	1
5	0	0	0
6	1	1	0
7	1	1	0
8	1	0	1

Decoding possible only after the system specific password is provided (sensitive to user PC and user's device configuration/id). On providing the password (password is device sensitive, applicable only for user PC) the unique bit pattern (Y0-Y7) is read from the ROM area. The shift registers are loaded with these bit patterns (Y0-Y7, different for all the 100 devices) and the 8 X 256 bit decoder is enabled for decoding of the 8-bit character. The receiver is able to read the original mail after it is XOR-ed with the bit pattern and decoded (as shown in table 5).

Coming back to the decoding mechanism, the bit sequence Z0-Z7 is transferred one by one with each clock pulse to the receiver's server.

As shown in figure 5, this bit sequence (Z0-Z7) is now decoded by first XOR-ing it with the bit-sequence (say Y0-Y7) same as C0-C7 stored in an 8-bit feedback register (G) at the receiver's end, with the last flip flop output being the input to the first one(as shown in table 4) and the result is stored in another 8-bit right shift register(T)(SERIALLY IN PARALLELY OUT). Even here for both the register we use a clock pulse with shift control generating octal word-time signal for 8-bits. Table 5 gives the ultimate binary value of the character to be decoded by the decoder.

After Y XOR Z (table 5), a 8 X 2^8 decoder is enabled and then the contents of 8-bit shift register having sequence T0-T7 is transferred to the input of the decoder. The information is decoded giving back the ASCII value -98

and thus giving back the corresponding character 'b'.

T7 T6 T5 T4 T3 T2 T1 T0
0 1 1 0 0 0 1 0

It's corresponding decimal value
[(2^7)*0] + [(2^6)*1] + [(2^5)*1] + [(2^4)*0] + [(2^3)*0] + [(2^2)*0] + [(2^1)*1] + [(2^0)*0] = 98---→ASCII value of 'b'

CLOCK PULSE	SHIFTING OF BITS WITH THE LAST OUTPUT BEING FED BACK AS THE INPUT	Output being XOR-ed with the received bit (Y0-Y7)
INITIALLY AT t=0	11101000	0
After t=1	01110100	0
After t=2	00111010	0
After t=3	00011101	1
After t=4	10001110	0
After t=5	01000111	1
After t=6	10100011	1
After t=7	11010001	1
After t=8	11101000	0

Above process repeated for all 100 characters transferred.

TABLE 4 : -THE SHIFTING OF BITS IN REGISTER 'G' AS SHOWN IN FIGURE 3 FOR 8 CLOCK PULSE

CLOCK PULSE	Z (Z0-Z7)	Y (Y0-Y7)	T=Y XOR Z (T0-T7)
1	0	0	0
2	1	0	1
3	0	0	0
4	1	1	0
5	0	0	0
6	0	1	1
7	0	1	1
8	1	1	0

TABLE 5 :- FOR THE LAST STEP WHERE ORIGINAL BIT SEQUENCE IS RETRIEVED

5 CALCULATIONS AND RESULTS

SET UP TIME FOR ONE FLIP FLOP=60ns
HOLD TIME FOR ONE FLIP FLOP=100ns
PROPAGATION DELAY (tp) = 200ns

Constructing a 10 x 10 dimension set up where 100 devices shown in figure 4 is connected in 2-d 10 *10 matrix format so as to transfer 800 bits at a time (figure 6). Considering that each feedback function to be unique; like in the example we have initially set the bit pattern to 3-bit right shift register to 000. System designer can follow any 3-bit combinations of 0's and 1's for the rest of the 99 devices.(say 001,101,110,111,010,100,011).This also work provided such different 8-bit decoding combinations are standardized at the receiver end's device.(in the example we have generated the bit pattern Y0-Y7: 00010111). Such different patterns must be provided to all the shift registers present at the receiver end. Even if any changes are made in the feedback bit sequence, receiver must be intimated about it and get the system modified as per the requirement (figure 5, figure 6 and figure 7).

The priority encoder takes about 2560ns to encode the 8-bit character.

1st bit transferred has got a delay of 1clock pulse (1000ns) + 1propagation delay(200ns).

2nd bit transferred has got a delay of 1clock pulse (1000ns) + 2propagation delay(400ns).

Therefore, 8th bit has got a delay of 1clock pulse and 8 propagation delay.

Total delay in transfer of 8-bit from the shift register shown figure 3 is 8*1000ns + (1+2+…….+8)*200ns

Td = (8000+7200) ns = 15200ns;

The linear feedback shift register's feedback function gives rise to delay of 70ns for computation of 1 function. Total of 7 functions take 490ns.

Shifting in the right shift register takes about 1600ns for 1 bit.

For 8-bit it takes 10800ns

XOR-ing of the data in the end takes another 10ns per bit

Total computation time = (2560+15200+11290+80)ns=29130ns

(around 29.13 micro seconds approx for transferring 800 bits);

The speed of transfer of email is dependent on the server used/speed of wireless network or broadband connection/file size etc.

After receiving the mail, only the sender's name is displayed. Rest of the data part is just sequence of 0's and 1's (no character is displayed, only bit sequence Z0-Z7).

After 2nd stage password check, the bit pattern (Y0-Y7) is loaded in the shift registers after being read from ROM (READ ONLY MEMEORY ,rate of data read 150Kb/sec, which is non volatile and data is not lost even after the power supply is cut off).

At the receiver end 100 bits received at a single clock pulse (common clock pulse guiding 100 decoding devices for the receiver's system).Time of about 1200 ns is taken for generating the bit sequence 100 bits at a single clock pulse and 10ns for XOR-ing with the incoming bit(Z0-Z7).(9680 ns for 800 bits). Decoding of 800 bits takes another 2560ns.So it can be estimated that a 25MB (megabytes) message may take 7.2825 min for encoding (figure 6) and approx. 5.78 min for decoding (figure 7).

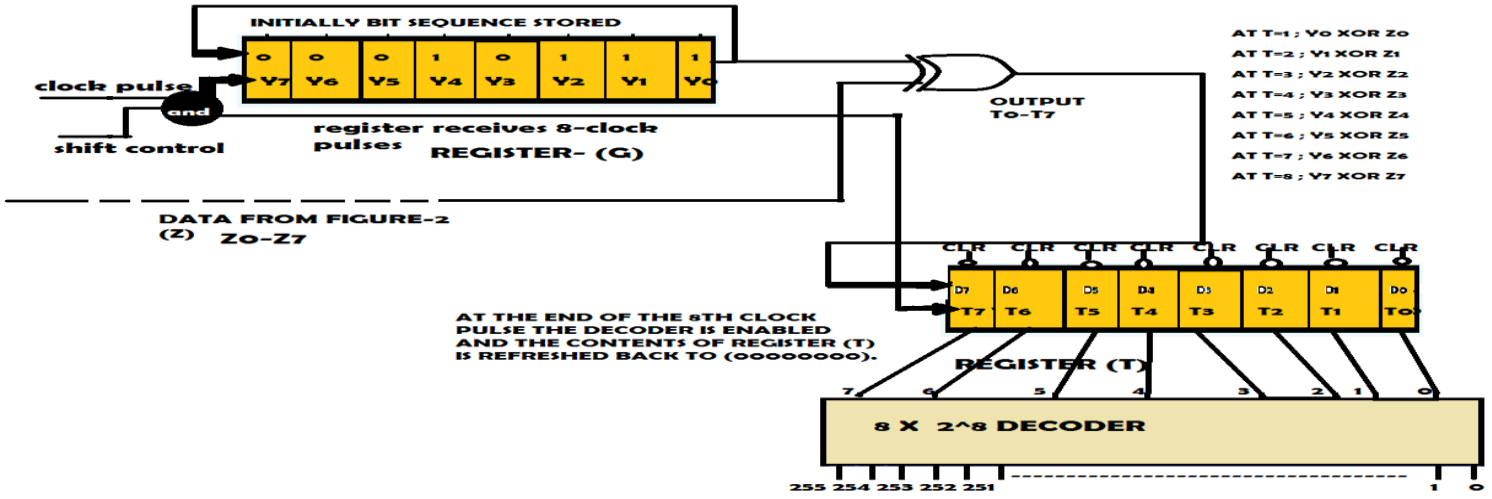

figure 5- DECODING OF ENCRYPTED DATA AT THE RECEIVER'S END

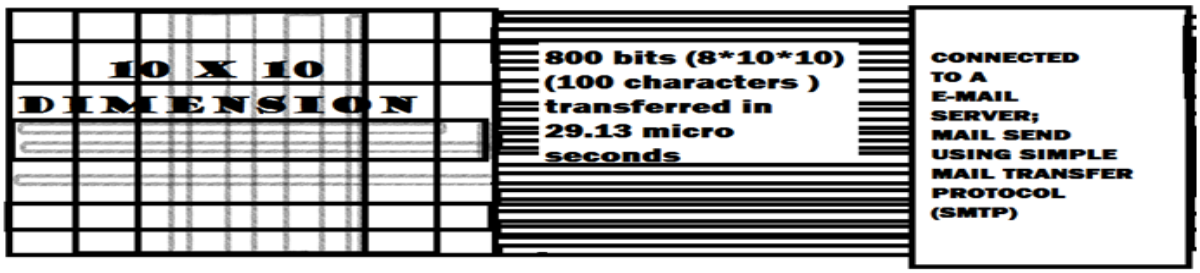

figure 6. SENDING OF THE ENCODED E-MAIL

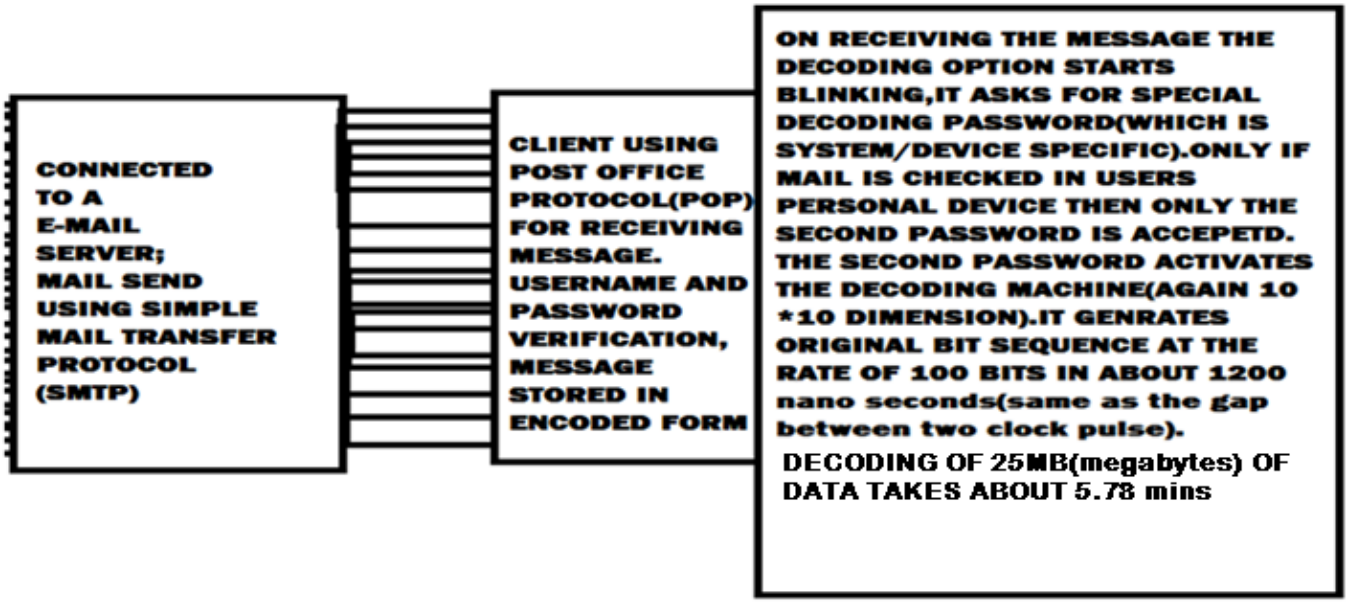

figure 7 : THE GENERATED BIT SEQUENCE RECEIVED IN THE E-MAIL IS DECODED ONLY AFTER UNDERGOING 2ND STAGE PASSWORD(DEVICE SENSITIVE) CHECK.

5 CONCLUSION

The following process does not come with 100% secured data transmission, but as number of bits being transferred increases from 8 to 16,32or 64, process of deriving the bit sequence for performing the decoding operation with the transmitted data becomes difficult for data attacker ,as it is a tedious process to match which combination of bits will work out. Even if the bit sequence is intercepted while transferring of data , discovering the correct combinations of 8-bit sequence (stored in ROM and to be XOR-ed with incoming bit sequence) for generating plain text from the cipher text is not at all feasible. Further it does not have a conventional polynomial feedback function but a mixture of various combinational operations. Here, one need

not worry about hacking of e-mail accounts as even if the password is known to the hacker, he or she will only be able to see the sender's name and sequence of 0's and 1's in text area. Unless and until the user works on his or her system, the original message can't be viewed. Once the message is viewed it is restored back to its encoded form instead of saving it in its decoded format (to view the original message later user must save it in his/her PC and avoid repeated decoding of same data). When the user operates via his account for decoding he is asked for device authentication prior to decoding. If the password(device sensitive) recognises the device id then only decoding process gets activated else it shows a failure message "decoding not possible" .Word time signal ensures shifting of bits in the shift register(in figure 5) thereby restoring the original bit sequence at the end of operation.

The statistics provided above is just based on theoretical calculations and needs experimentation to compute the data transfer time and whether any further modifications could be added (like data compression)for reducing the complexity of algorithm. As of now we can't claim it to be a 100% secured data transfer technique and can be applied only to text files attachments, or text mails. Picture files (graphics, RGB files) /AV files will not follow the above process while encoding.

ACKNOWLEDGEMENT

The authors are very grateful to Mr. Bibhash Sen , Assistant Professor, Department of Computer Science and Engineering, for his support during preparation of the paper and his willingness to share ideas and helpful suggestions throughout dissertation.

The authors would like to express their gratitude and appreciation to their dear parents and all their teachers who taught everything since their childhood.

REFERENCES

[1] *Digital Logic and Computer Design*, M.MORRIS MANO, California State University, Los Angeles,4th edition, 2008
[2] *Lessons in Electric Circuits-volume IV*, file:///I:/Lessons%20In%20Electric%20Circuits%20--%20Volume%20IV%20(Digital)%20-%20Chapter%2012.htm
[3] *Feedback Shift Register Sequences* , by HONG-YEOP SONG, Department of Electrical and Electronics Engineering, Yonsei University, Seoul, 120-749, Korea
[4] file:///I:/Linear%20feedback%20shift%20register%20-%20Wikipedia,%20the%20free%20encyclopedia.htm
[5] file:///I:/Linear%20Feedback%20Shift%20Registers.htm
[6] *Design and Analysis of Stream Cipher-II*, Lecturer- SOURADYUTI PAUL, Computer Security and Industrial Cryptography(COSIC), Katholieke Universiteit Leuven, Belgium
[7] file:///I:/Stream%20cipher%20-%20Wikipedia,%20the%20free%20encyclopedia.htm
[8] *Stream Ciphers-ppt* , CS 519, Cryptography and Network Security, Instructor: ALI AYDIN SELCUK
[9] *Analysis and Design,* on lfsr based stream cipher, PATRIK EKDAHL, Department of Information Technology, Lund University,2003
[10] *Speech Encryption and Decryption Using Linear Feedback Shift Register (LFSR),* TIN LAI WIN, and NANT CHRISTINA KYAW
[11] *All About Circuits,* Ring Counter file:///I:/Ring%20counters%20%20%20SHIFT%20REGISTERS.htm