FPGA Implementation of Dynamic Key Generation To Enhance Des Algorithm Securities

Akhilesh Gautam
Electronics and Communication Engineering Department
Shri Vaishnav Institute of Technology and Science
Indore, Madhaya Pradesh

Prof. Preet Jain
Electronics and Communication Engineering Department
Shri Vaishnav Institute of Technology and Science
Indore, Madhaya Pradesh

***Abstract*— In cryptography, Key play an important role. For higher security of any secure communication, the secrecy of key is essential. The objective of this project is to demonstrate FPGA implementation of Dynamic key generation for "Data Encryption Standard (DES)" algorithm which makes DES strong. In this paper, we develop a dynamic key generation unit to enhance the DES security. The developed system has two main component, first is key generated by LFSR and second one is key generated using Chaotic encryption.**

The proposed dynamic key generation unit is independent of DES algorithm which makes its more effective. This design also cancelled the weak keys and generate strong key.

The result of this design has higher resistance against brute-force attack.

Keywords- DES, Dynamic Key Generator, FPGA, LFSR, Chaotic Encryption.

I. BACKROUND

In any cryptographic algorithms, the security of encrypted data is not only dependent on the secrecy of its algorithm but also more dependent on the secrecy of its key. Key generation plays an important role in cryptography. Key generation is the hardest job in cryptography. Because once the key is exposed, the whole encryption algorithm doesn't work. In this paper we implemented a hardware encryption of DES and its key generation unit using FPGA

.

DES is one of the best known cryptographic algorithms, and has been used since when it introduces in 1976 and is still used today despite the fact that it doesn't offer a sufficient level of security [1]. The DES algorithm is symmetric key cipher that uses the same binary key both to encrypt and decrypt data blocks. DES operates on 64-bit "plaintext" data blocks, processing them under the control of a 64-bit key to produce 64 bits of encrypted cipher text. DES uses a sequence of operations, including several substitution and permutation primitives, to encrypt a data block.

This paper presents a dynamic key generation unit to enhance security of conventional Des algorithm. The key can be reconfigured by this unit. The dynamic key generation unit can be adopted to increase confusion of the key and secure the protection of DES algorithm.

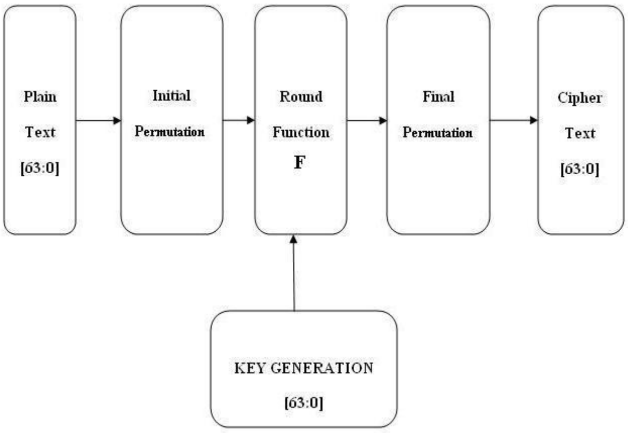
FPGA (Field Programmable Gate Array) technology is the best choice of implementation of the DES algorithmic and proposed design due to its flexibility, physical safety and higher speed. FPGA is a predesigned reconfigurable IC. It can be reconfigured any number of times according to the speciation of design. The FPGA configuration is generally defined using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). The HDLs are VHDL and Verilog. We prefer VHDL for programming because of its widely in use [7]. The function simulation and test analysis based on the design is verified, and the result is satisfied.

II. ALGORITHM DESCRIPTION

A. Data Encryption Standard algorithm Theory

DES is a symmetric key modern block cipher algorithm. It encrypts 64-bit block of plain text or decrypts 64-bit cipher text. For encryption and decryption, it uses 64-bit key. A block diagram of DES is shown in **Fig. 1.1.**

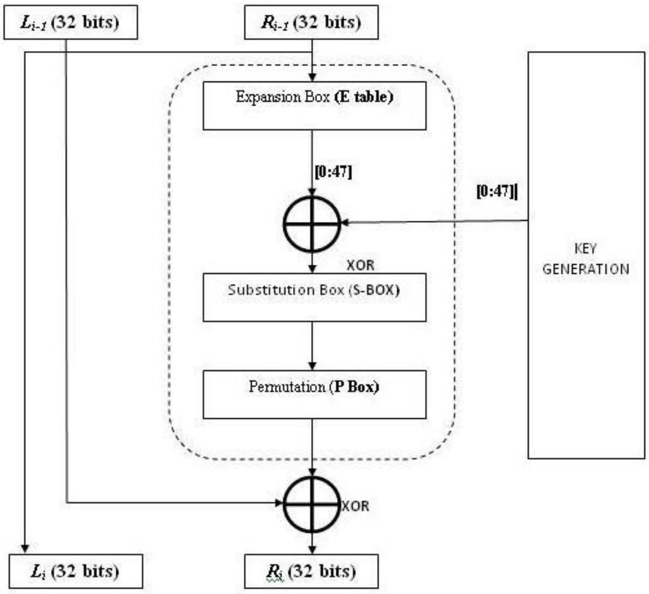
As we already know that DES take input of 64-bit block of plain text, after that these 64-bit of plain text is rearranged by initial permutation block, this block permuted input bits in predefined manners. The purpose of this block is to introduce diffusion in plain text bits Diffusion means hiding the relationship between plain text and cipher text.



BLOCK DIAGRAM OF DES ALGORITHM

Fig. 1.1

After initial permutation the 64-bit data comes in round function block. DES has 16 rounds. In each round, a round function performed..The round function is heart of DES algorithm, the block diagram of round function is shown in Fig. 1.2.



BLOCK DIAGRAM OF FEISTEL (F) FUNCTION

Fig. 1.2

As shown in fig 1.2, in a round function, the output of **IP** block which is 64 bit wide splits into two equal parts. First is Right half (R_0) and second is left half (L_0), each 32 bits in length.

As shown in figure fig.1.2, the 32-bit right half of the plaintext $\boldsymbol{R_{i-1}}$ is expanded to 48-bits by expansion permutation block and then XORed with a 48-bit sub-key $\boldsymbol{K_i}$. The result is then fed into eight substitution boxes (**S**-boxes), which transforms the 48-bit input to a 32-bit output. Finally, a straight permutation (**P**-permutation) is performed, the output of which is XORed with the left half $\boldsymbol{L_{i-1}}$ to obtain the new right half $\boldsymbol{R_i}$. The right half $\boldsymbol{R_{i-1}}$ becomes the new left half $\boldsymbol{L_i}$ [3].

The S-box is the critical part of the DES algorithm. It realizes the non-linear transformations.

The First step of key generation is to remove the parity check bits in the 64-bit key. Every eighth bit is used for parity checking, leaving 56-bits. The parity bits are 8, 16, 24, 32, 40, 48, 56 and 64 bit. Now a different 48-bit sub-key is generated for each of the 16 rounds of DES. The sub keys are determined by dividing the 56-bits into two 28-bit lengths of data. Then both halves are shifted left by either one or two bits depending on the round number. In rounds 1, 2, 9 and 16 of the DES algorithm the halves are shifted one position to the left and for all other rounds two positions to the left.

In Fig. 1.1, it shows that key generator is independent of DES algorithm computation, which offers the potential and convenience for key configuration.

B. Linear Feedback Shift Register

LFSR is an acronym for Linear Feedback Shift Register. Due to LFSR is easy to constructed and implemented by software and hardware, so that it can be used to as a Good key Stream generator. A LFSR consist a shift register and a linear feedback function of its previous states. As shown in Fig. 1.3.

The shift register is sequence of M flip flops, $\mathbf{B_M}$ to $\mathbf{B_{M-1}}$, where **e**ach flip flop holds a single bit. The flip flops are initialized to an M-bit word called the **Seed.** As shown in Fig 1.3, $\mathbf{B_M}$ is a linear function of $\mathbf{B_0}$, $\mathbf{B_1}$, $\mathbf{B_2}$,**…**,$\mathbf{B_{M-1}}$ [1].

Shift register can be divided according to its type of inputs and outputs. For example serial inputs and parallel outputs or parallel inputs and serial inputs.

LFSR has broad area of applications. The main domain of LFSR applications include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters and whitening sequences [7].

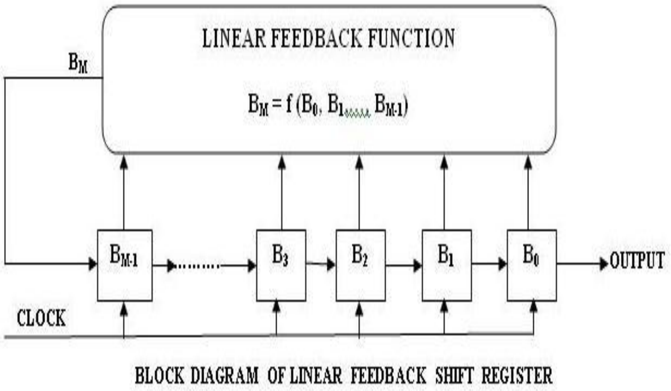

LINEAR FEEDBACK FUNCTION
$B_M = f(B_0, B_1, \ldots, B_{M-1})$
B_M
B_{M-1}
B_3
B_2
B_1
B_0
OUTPUT
CLOCK
BLOCK DIAGRAM OF LINEAR FEEDBACK SHIFT REGISTER

Fig. 1.3

C. *Chaotic Encryption.*

In recent years, chaotic systems are more researched in the field of key generation of cryptography. In general, “chaos” means “a state of disorder”. However, in chaos theory means a nonlinear dynamic behavior of the law control [5].

For a dynamical system to be classified as chaotic, it must have the following properties:

- It must be sensitive to initial conditions;
- It must be iterative unrepeatability

One-dimensional Logistic map is a very simple chaotic map from the point view of mathematical form. But this system has an extremely complex dynamics, wide applications in the field of secure communication, so that the Logistic map can be used for “Dynamic key generation unit” [4].

Logistic model is one of the chaotic models; its equation is as follows:

$$Y_{N+1} = \mu \times Y_N \, (1 - Y_N) \quad (1)$$

Where (N = 0, 1, 2…)

And the initial value $Y_N \in (0,1)$.

The research of chaotic dynamical systems points out that the logistic map gradually reaches to chaos from times bifurcation phenomena when $3 \leq \mu \leq 4$. That is, $\{Y_N, N=1,2,3,\ldots\}$, which is produced by Logistic map. It is non-periodic and convergence under initial condition.

In theory chaotic sequence is not pseudo-random, but truly stochastic, so key generator with high intension can be constructed using these characteristics [2].

III. DYNAMIC KEY GENERATION

A. *Proposed Design*

According to kerckhoff’s principle, one should always assume that the adversary. The resistance of the cipher to attack must be based only on the secrecy of the key. In other words, guessing the key should be so difficult that there is no need to hide the encryption/decryption algorithms, but in the case of DES algorithm, its main weakness is its 56-bit key. Using 56- bit key, there are 2^{56} possible keys available, now a day which are easily crack by “brute force attacks”. “Brute-force” means in which involves trying all 2^{56} keys. Apart from this out of 2^{56} keys, 4 are weak key, 12 are semi weak and 48 are possible weak key [1].

If we increase number of possible key generation way more than 2^{56}, then we can enhance security in DES.

To increase number of ways of key generation more than 2^{56}, we proposed a dynamic key generation unit in DES.

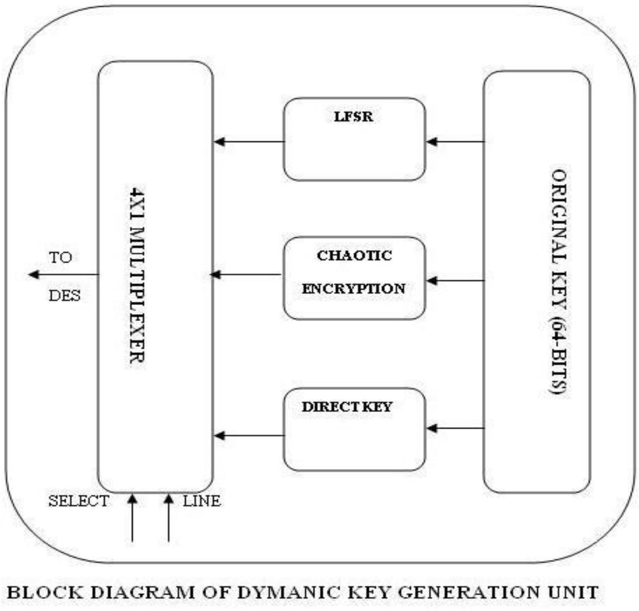


Fig 3.1

As shown in **Fig. 3.1**, the dynamic key generation unit consists three blocks named LFSR, chaotic encryption, and direct key respectively.

For example if a user generates a key for DES, he has 2^{56} ways but after key shifted through LFSR there are also 2^{56} ways available (depending upon feedback function), so that finally user has 2^{112} way. For make more confusion in key generation, the key can be generated by any of the above methods.

B. Hardware Implementation Structure

Form the point of view of hardware implementation this project, we have divided this into two part. The first part is conventional DES algorithm part and another part is key generation part. A control unit also initiate here to control round of DES and encryption/decryption mode. The conventional DES algorithm part has two main set of input signal: plain text *DIN(63:0)* and *KEY(63:0);* output signal cipher text *DOUT(63:0);* and a valid input signal MODE. For mode signal 0 indicates Encryption and 1 Decryption indicates. In this design, an input signal *SEL(1:0)* is added to realize the dynamic key configuration; when the value of *SEL(1:0)*is 2'b01 or 2'b10 , the input signal *KEY(63:0*) is directly sent to the key generation module of DES module; when it takes 2'b00, the key *KEY(63:0*) is firstly sent to the LFSR module, and a new sequence is gotten, then the sequence is sent to the key generation module; when it takes 2'b11, the key *KEY(63:0)* is firstly sent to the Logistic module, and a chaotic sequence is gotten as new key to participate in the following encryption operations.

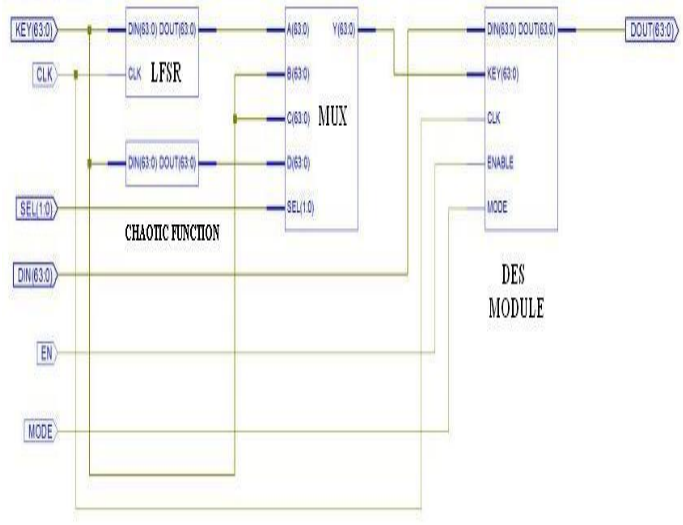


Fig.3.2

As we can see that, the LFSR module and Logistic (chaotic) module cannot simultaneously work. Because they are input of multiplexer and selection of these module depend upon *SEL(1:0)* value. The value of *SEL(1:0)* have to known for both end of communication which enhance security of key because it is unknown for attackers. Selecting the corresponding encryption algorithm by selecting *SEL(1:0),* which can simplify the circuit, reduce the numbers of the gates on chip, and reduce integration of chip. Then it can achieve the purpose of reducing power consumption. Its logical structure is shown in Fig. 3.2.

DES encryption algorithm as the core of this design has two characteristics: several rounds of the key transform round function and key/data operation with round function. The round function plays an important role in DES algorithm and it use 16 times in DES so that required a hardware unit to implement it. For implementation round function we used time-sharing multiplexing of a hardware implementation of round function, and call a round function hardware copy repeatedly [6].

Because the DES encryption algorithm is implemented by calling the hardware structure 16 times again and again so that hardware cost is reduced greatly. Of course the chip performance is a little reduced. For make this design fast and efficient, we have used behavioral modeling.

IV. SIMULATION AND RESULTS

All of the algorithms are implemented with VHDL and simulated in RTL level with testbench under Xilinx ISE Simulator environment, and the timing simulation waveform is shown in **Fig. 4.1, Fig. 4.2, and Fig. 4.3** respectively.

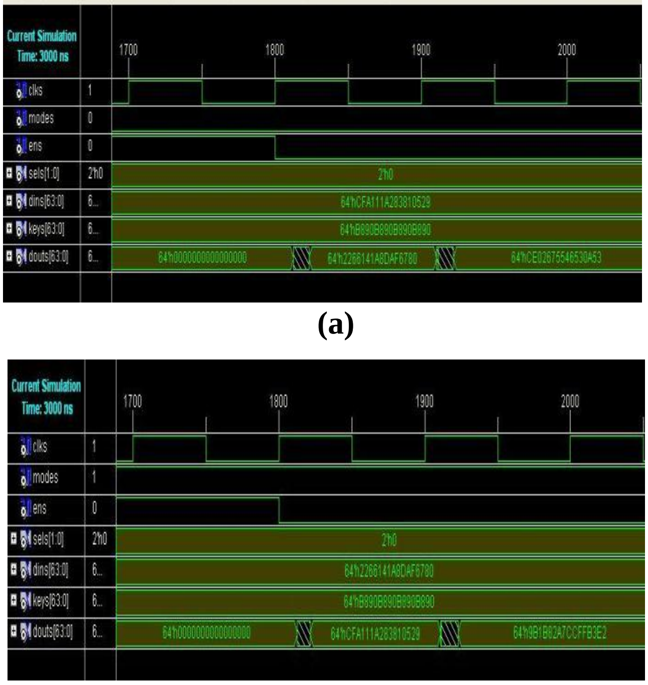
Each Following simulation figures contain two part named (a) and (b): (a) part shows when DES is in encryption mode. In other words when value of *MODE* is "0" and (b) shows when DES is in decryption mode or In other words when value of *MODE* is "1". The output of figures *DOUT[63:0]* contains three 64-bits long value. The first value indicates initial value, the second value show that final output and third one show that garbage value.

For Encryption we used input value *DINS[63:0]* 64'HCFA111A283810529 and *KEYS[63:0]* 64'hB890B890B890B890**.**

For decryption, we used input value as its Encryption output value and same keys.

Select signals *SEL[1:0]* indicates selection of key generation module. When value of *SEL[1:0]* is "00" the LFSR module selected, When value of *SEL[1:0*] is "01" and "10" direct key generated and When value of *SEL[1:0]* is "11" key generated by CHAOTIC Encryption.

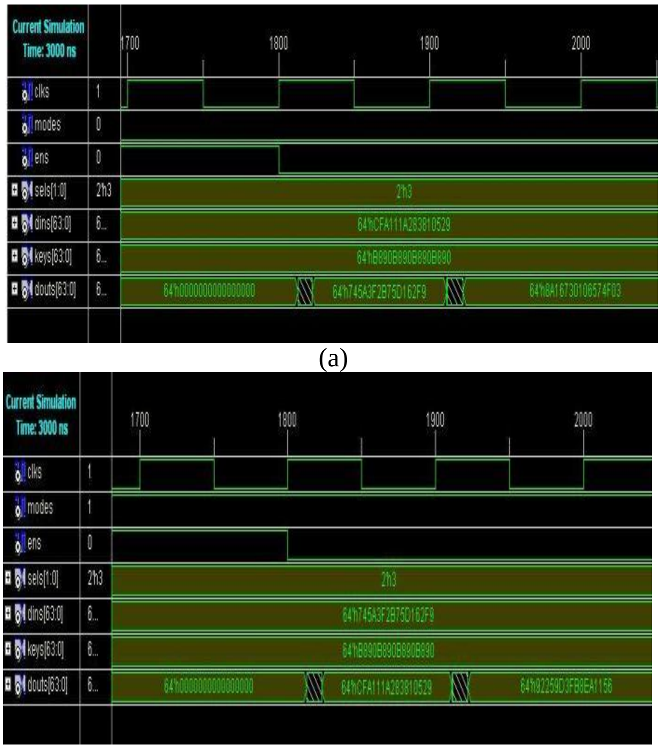
Key generated by LFSR Module



(a)

(b)
Fig. 4.1

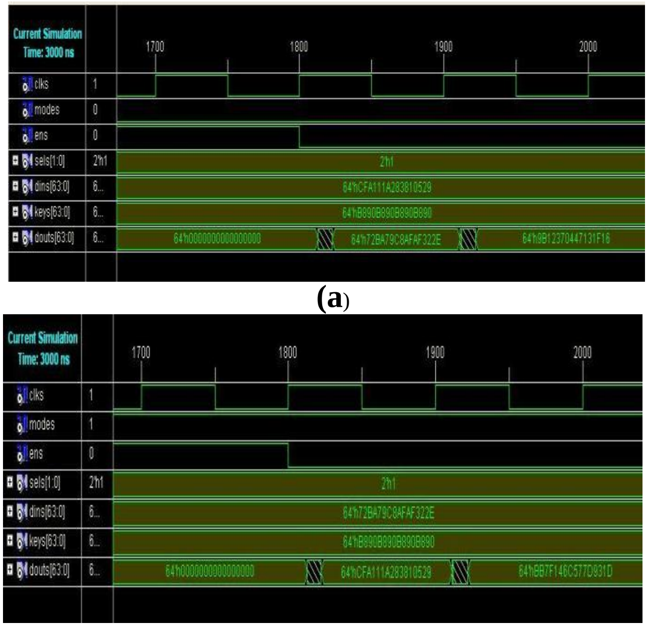
Key generated by logistic or chaotic function



(a)

(b)
Fig. 4.2

When dynamic key generated by direct key



(**a**)

(b)

Fig. 4.3

V. CONCLUSIONS

The security of any encryption algorithms is not only dependent on the secrecy of its algorithm, but also more dependent on the security of the key used. The whole of the conventional DES algorithm is analyzed in this paper. The reasonable DES encryption algorithm design methods and approaches based on dynamic key generator are put forward. And it has been simulated and verified in the FPGA platform. The data can be encrypted and decrypted continuously with higher security. The results indicate that the developed system completely meets the timing requirements and achieves the design purpose.

REFERENCES

1. Behrouz A. Forouzan, “Cryptography and Network Security”, Special Indian Edition., TMH: New Delhi, 2007.
2. Chen Zhuo, Zhang zhengwen, Jiang Nan, “A Session Key Generator Based on Chaotic Sequence.” International Conference on Computer Science and Software Engineering, 2008, DOI 10.1109/CSSE.2008.833.
3. William Stallings, “Cryptography and Network Security- Principles Practice”, Fifth Edition, Prentice Hall, 2006.
4. Ji Yao, Hongbo Kang, “FPGA Implementation of Dynamic Key Management for DES Algorithm” International Conference on Electronics & Mechanical Engineering and Information Technology, Dated 12-14 Aug 2011.
5. Sivaprakasam S and Shore K A, “Message encoding and Decoding using chaotic external-cavity diode lasers,” IEEE Journal of Quantum Electronics, 2000, 36(1):35- 39.
6. Ke Wang, “An encrypt and decrypt algorithm Implementation on FPGAs”, 2009 Fifth International Conference on Semantics, Knowledge and Grid, DOI 10.1109/SKG.2009.74.
7. Amit Kumar Panda, Praveena Rajput, Bhawna Shukla,“FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial using VHDL”, 2012 International Conference on Communication
8. Systems and Network Technologies, DOI 10.1109/CSNT.2012.168.