Effect of newly developed data security algorithm on the 128-bits plaintext and study of resistance to ciphertext attacks with maximum combinational path delay using VHDL

Paresh Kumar Pasayat[1], Kalpataru Sethi[2]

[1]Assistant Professor, Dept. of ETC Engineering, IGIT,Odisha, India
[2]PG student, Dept. of ETC Engineering, IGIT,Odisha, India

---***---

Abstract – *The proposed paper focuses mainly on the hiding of information from the unauthorized access and the novel approach adopted so as to prove the robustness of the data security algorithms. The information consists of 128-bit plaintext which needs to be protected from the hacker by using a newly developed data security algorithm. The algorithm protects the data by performing various operations on the plaintext and the chip codes and obtains a 128-bit ciphertext. The different operations have been achieved with the help of bit splitter unit, bit append unit, xor unit and a dependent functional block. The original data can be retrieved at the receiver end by using reverse cipher unit. The proposed work can be implemented in the banking sector, telecommunication sector and military sector etc.*

Key Words: Plaintext, Ciphertext, Bit Splitter, Bit ppend, Cipher, Functional Block.

1. INTRODUCTION

Due to the lack of confidentiality, integrity and authentication of the information being used by the different organizations, the researchers are playing an important role for inventing various data security algorithms to protect the information from the unauthorized access. The proposed paper aims to protect 128-bit data by introducing a newly developed data security algorithm. First, 128-bit plaintext is converted into 128-bit ciphertext using two 32-bit chip codes and cipher algorithm. Similarly, the 128-bit ciphertext is converted into 128-bit plaintext by using reverse cipher algorithm.

1.1 Project Model

The project describes the flow chart for the proposed project work. Each number in the model signifies the no. of bit in the input and output of each unit. The diagrammatic representation of the proposed work is given as follows:

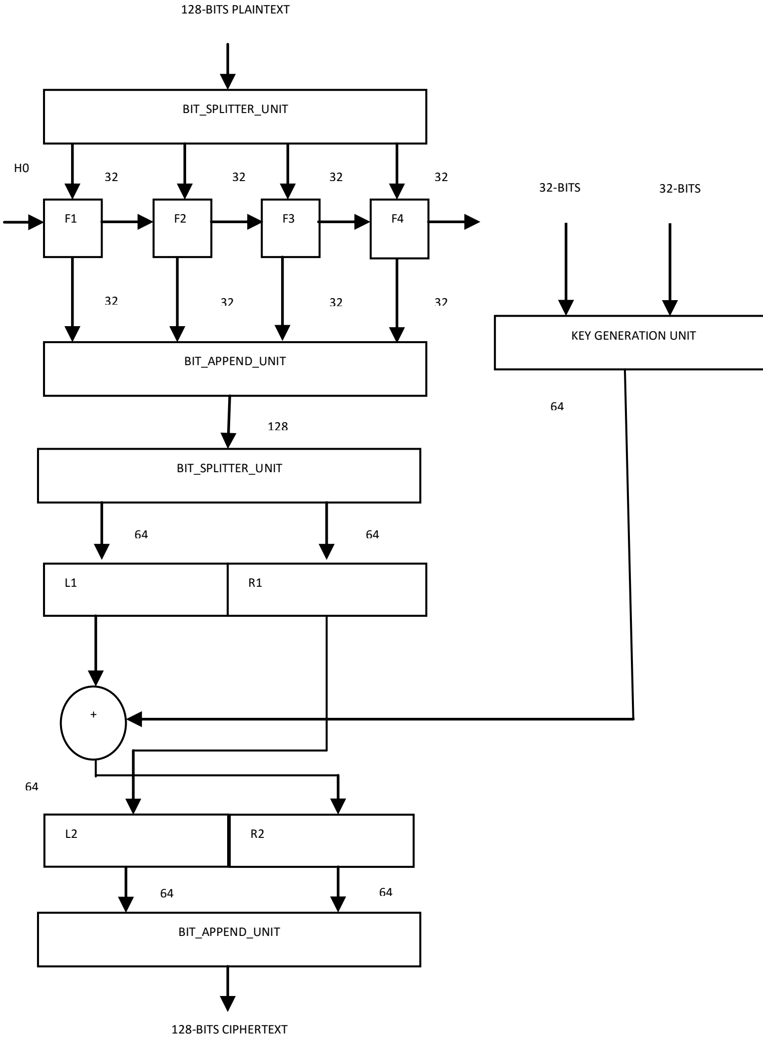


Fig 1: Project Model

2. LOGIC USED IN THE PROPOSED DESIGN

The logic used in the proposed design has been described in different steps as follow:

2.1 ENCIPHERMENT ALGORITHM:

Step 1: First, 128-bits Original data also know as plaintext is fed to the input of the bit splitter unit which divides the data into four half each having equal no. of bits i.e. 32-bits.

Step 2: The outputs of bit splitter unit and the hash value are given to the inputs of the new functional unit (F1) which produces two outputs each having 32-bits (one output given for the bit append unit and another input to the input of the new functional unit). This process is repeated for next three nos. of Fi where i=2 to 4.The hash value of next Fi is the output of the previous Fi.

Step 3: The outputs of Fi are appended with the help of bit append unit which produces 128-bits output.

Step 4: Two 32-bits cipher keys are appended together to produce 64-bits key output for the Feistel Cipher Unit.

Step 5: Feistel Cipher Unit divides the output of bit append unit into two halfs (L1 & R1) each having 64-bits. Then, the XOR operation is performed between L1 and the key. The output of XOR unit and R1 are swapped using the swapper unit. Then, the outputs of the swapper unit is appended using bit append unit which produces 128-bits Ciphertext output. This ciphertext output is the desired encrypted data to be transmitted from the transmitter to the receiver through wire / wireless medium.

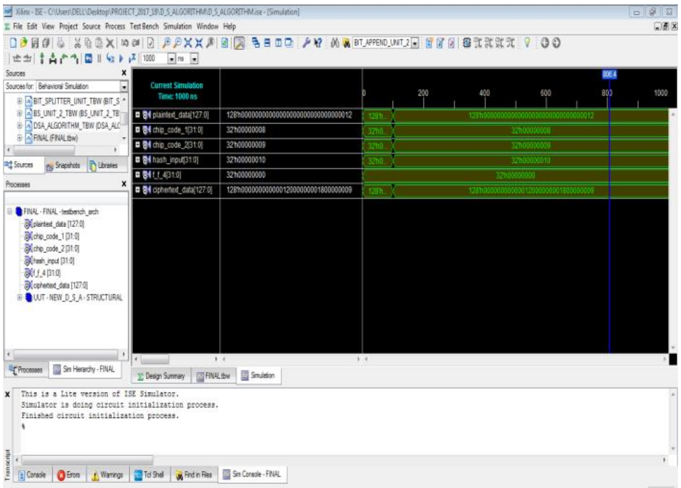
2.2 DECIPHERMENT ALGORITHM:

The algorithm for the decryption process can be written in the reverse order of the encryption algorithm.

3. SIMULATION RESULT AND DISCUSSION

The VHDL code of the proposed work has been simulated using Xilinx ISE 9.2i software and the desired results have been obtained.

The simulation result of the encipherment process is given as follows:

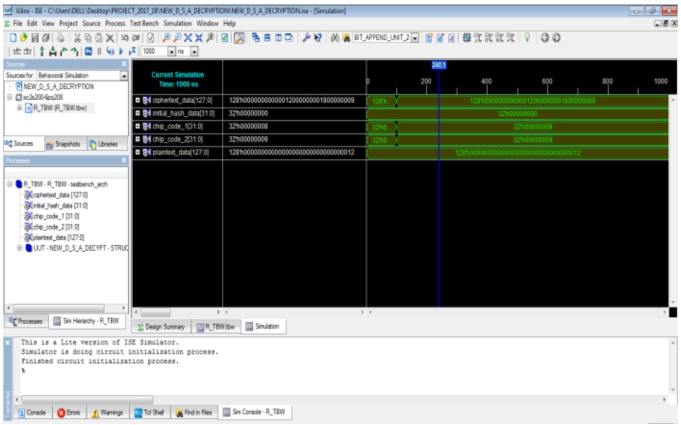

Xilinx - ISE - C:\Users\DELL\Desktop\PROJECT_2017_18\D_S_ALGORITHM\D_S_ALGORITHM.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
BIT_APPEND_UNIT_2
Sources
Sources for: Behavioral Simulation
BIT_SPLITTER_UNIT_TBW (BIT_S
BS_UNIT_2_TBW (BS_UNIT_2_TB
DSA_ALGORITHM_TBW (DSA_ALC
FINAL (FINAL.tbw)
Sources Snapshots Libraries
Processes
FINAL - FINAL - testbench_arch
plaintext_data [127:0]
chip_code_1 [31:0]
chip_code_2 [31:0]
hash_input [31:0]
f_f_4 [31:0]
ciphertext_data [127:0]
UUT - NEW_D_S_A - STRUCTURAL
Processes Sim Hierarchy - FINAL
Current Simulation Time: 1000 ns
0 200 400 600 800 1000
Design Summary FINAL.tbw Simulation
Console Errors Warnings Tcl Shell Find in Files Sim Console - FINAL
Transcript

```
plaintext_data[127:0]    128'h00000000000000000000000000000012
chip_code_1[31:0]        32'h00000008
chip_code_2[31:0]        32'h00000009
hash_input[31:0]         32'h00000010
f_f_4[31:0]              32'h00000000
ciphertext_data[127:0]   128'h00000000000001200000001800000009
```

```
This is a Lite version of ISE Simulator.
Simulator is doing circuit initialization process.
Finished circuit initialization process.
%
```

Fig 2: Simulation result of the encipherment process

The simulation result of the decipherment process is given as follows:


Xilinx - ISE - C:\Users\DELL\Desktop\PROJECT_2017_18\NEW_D_S_A_DECRYPTION\NEW_D_S_A_DECRYPTION.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
BIT_APPEND_UNIT_2
Sources
Sources for: Behavioral Simulation
NEW_D_S_A_DECRYPTION
xc2s200-6pq208
R_TBW (R_TBW.tbw)
Sources Snapshots Libraries
Processes
R_TBW - R_TBW - testbench_arch
ciphertext_data [127:0]
initial_hash_data [31:0]
chip_code_1 [31:0]
chip_code_2 [31:0]
plaintext_data [127:0]
UUT - NEW_D_S_A_DECYPT - STRUC
Processes Sim Hierarchy - R_TBW
Current Simulation Time: 1000 ns
0 200 400 600 800 1000
Design Summary R_TBW.tbw Simulation
Console Errors Warnings Tcl Shell Find in Files Sim Console - R_TBW
Transcript

```
ciphertext_data[127:0]     128'h00000000000001200000001800000009
initial_hash_data[31:0]    32'h00000000
chip_code_1[31:0]          32'h00000008
chip_code_2[31:0]          32'h00000009
plaintext_data[127:0]      128'h00000000000000000000000000000012
```

```
This is a Lite version of ISE Simulator.
Simulator is doing circuit initialization process.
Finished circuit initialization process.
%
```

Fig 3: Simulation result of the decipherment process

4. CONCLUSION

After doing the proposed work, it is concluded that the work is best suited in the field of data security to provide protection to the 128-bits original data from unauthorized access. It is resistant to the brute-force attack, timing attack, snooping attack, pattern attack, statistical attack which makes the algorithm more robust. The combinational path delay of the time required to convert 128-bits plaintext into 128-bits ciphertext is 8.063ns which obtained from the Xilinx software.

REFERENCES

[1] Dr. Sandeep Tayal, Dr. Nipin Gupta, Dr. Pankaj Gupta, Deepak Goyal, Monika Goyal, "A Review paper on Network Security and Cryptography", Advances in Computational Sciences and Technology, Volume 10, Number 5,pp. 763-770,2017.

[2] J. G. Pandey,Aanchal Gurawa, Heena Nehra,A. Karmakar, "An efficient VLSI architecture for data encryption standard and its FPGA implementation",VLSI SATA,IEEE International Conference,pp.1-5,2016.

[3] W.Stallings,"Cryptography and Network Security", 2nd Edition, Prentice Hall.

[4] Douglas L. Perry. "VHDL Programming by Examples", TMH.

[5] Soufiane Oukili,Seddik Bri,"FPGA implementation of Data Encryption Standard using time variable permutations", International Conference on Microelectronics (ICM),IEEE,pp.126-129,2015.

[6] Ramadhan J. Mstafa; Khaled M. Elleithy, "A highly secure video steganography using Hamming code (7, 4)", Systems, Applications and Technology Conference (LISAT), IEEE Conference,pp.1-6,2014.

[7] Ravikumar M.Raypure, Prof. Vinay Keswani, "Implementation For Data Hiding Using Visual Cryptography", IRJET, Volume: 04, Issue: 07, 2017.