2022 SSCS "PICO" Open-Source Chipathon
Pre-Layout Documentation

Group: Pakistan 2

Project Name: Encrypted LSB Steganography Implementation with AES-128 accelerator

Group Members: Abdul Moiz Sheikh, Ruhma Rizwan, Ali Sabir, Osama Liaqat, Mubashir Saleem

Date: 10-01-2022

Contents

Pre-layout Documentation

Project Details

1. Abstract

Various techniques are utilized today to secure sensitive data from unwanted access; the most effective are encryption and steganography. Steganography deals with concealing confidential information inside non-secret information using a mask to make it non-distinguishable. AES-128 is one of the most popular and strongest encryption ciphers in use nowadays. Dedicated hardware implementations of these techniques provide more security and higher performance compared to software implementations.

In this project, we implement a high-speed hardware architecture of a combined unit for AES and Steganography of standard 128-bit input data. The proposed architecture implementation on 130 nm pdk provides a throughput higher than 800Mbps and power consumption of less than 10mW. Moreover, the proposed architecture achieves NIST compliant accuracy. The AES algorithm is implemented using a tradeoff in reference [1], [2], and [3] for maximum power, area efficiency and throughput.

2. Motivation

Cyber-security has become vital for secure systems worldwide, providing encryption for millions of sensitive financial, government, and private transactions daily. Security services in various environments utilize popular techniques of cryptographic algorithms and steganography for the encryption and masking of sensitive data. With the surge of information and the increasing need for secure communication in real-time, low cost, higher speeds, and low power consumption have become the key requirements in such services. Software implementations of these services are much more time-consuming and less secure as compared to dedicated hardware implementations. Therefore, we propose a high-speed hardware architecture for the implementation of encryption and steganography of data that provides higher performance, throughput, and lower power consumption.

3. Background

Encryption and Steganography are two popular and predominant techniques utilized for the security of classified data.

- *Encryption*

Encryption transforms the confidential information (payload) into gibberish (cipher text) with the help of a secret password (key) and this gibberish is then transmitted through the channel. The intended receiver can decode the cipher text if they have the appropriate key to retrieve the actual data. AES-128 is one of the most commonly used encryption ciphers today and is one of the strongest. It is an example of a block

and symmetric cipher which uses a single key for encryption and decryption and operates with a larger, pre-saved chunk of data.

- *Steganography*

Steganography, on the other hand, is the art of concealing secret information (payload) inside the non-secret information (or cover) such that the attacker is unable to distinguish and extract the secret message from the cover. The cover can be any medium, text, image, video, etc. In LSB steganography, the LSB bit plane/s of the cover is modified and secret message bits are embedded in these planes. The cover is transmitted and the intended receiver; knowing the areas in the cover which conceal information; can unmask the information.

This payload can be encrypted using AES-128, followed by steganography to provide additional security.

The above-stated operations are performed quite frequently using the software stack. However due to the ever-increasing flow of information and the increasing need for secure communication; it has become necessary to perform these operations faster and in real-time. The software stack implementation, on the other hand, is relatively slower due to the inherent use of abstraction layers.

4. Proposed Solution

In this project, we implement a hardware implementation of AES encryption and steganography which provides higher throughput and higher area efficiency as it is one of the main concerns in ASIC design. We propose an overall system with two sub-components; an AES-128 accelerator and a steganography unit.

The secret payload is inputted to the system and is acquired in 128-bit chunks which are standard for AES-128. The AES core encrypts this payload with the help of the 128-bit key also given as an input to the system. Different components of the AES are optimized for area and power efficiency. The 'SubBytes' step requires a lot of memory to be allocated which can be optimized as explained in [1]. It is implemented using combinational logic instead of lookup tables to eliminate the impact of its delay. The AES block also implements a tradeoff between fully parallel architecture with higher speed and throughput but more area as in [2], and a more compact higher latency design as in [3]. The AES core implements a masked version of the payload to protect it from any side-channel attacks or power analysis attacks.

The resulting output of the AES block is fed into the steganography block which accepts it as a payload and masks it in the cover data using the LSB steganography approach. The cover is taken as an input, a few bytes at a time, and is processed and transmitted to the output. Consequently, the incoming cover data stream is not required to be pre-saved and the results only suffer the delay of the AES block and the negligible delay of embedding payload into the cover.

The proposed design is implemented in an asynchronous manner for speed of operation and accuracy. The system has a few flags which are set and cleared during the time of operation including 'EB'

(Encryption Busy), SB (Steganography Busy), etc. which facilitate the operation of the system. The two sub-blocks AES core and Steganography core work in parallel in a pipelined manner.

Implementation

1. AES-128 Encryption

The AES block receives the key, input data, and initialization vector as input and gives the encrypted data payload as output along with flags denoting the end of encryption. We are using the Cipher Block Chaining (CBC) mode of operation in order to achieve more randomness. The Counter, initially 0; retains a value of 1 after the first change in output. The output is received after 23 cycles. The output changes despite the input remaining the same testament to the change in Initialization Vector (IV). Figure 1 shows the flow of the AES-128 Encryption operation.

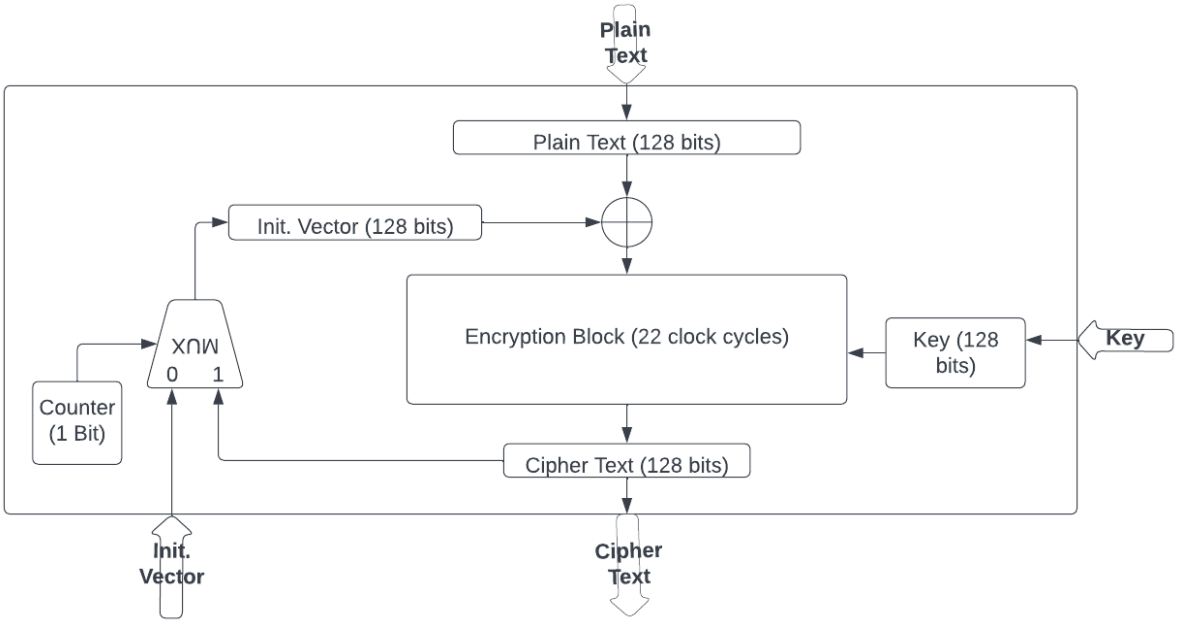

Figure 1: Flow of AES128 Encryption with CBC operation

The simulation results are as shown in Figure 2.

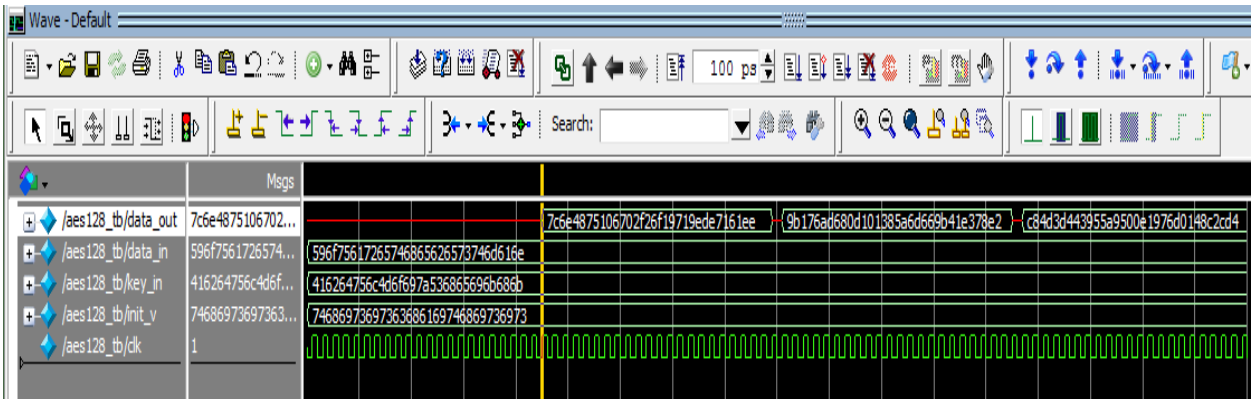

Figure 2: Simulation results of aes128

2. Steganography

The Steganography receives the 128 bit payload and 8-bit cover data as input and sends the final encrypted cum stegnographic coded data along with flags denoting the end of steganography. We are using the last two bit planes of the cover data to embed the payload for achieving higher throughput. The embedding sequence is Most Significant bits (MSB's) first i.e. in the first byte of cover data, the two MSB's are embedded and so on until the last two Least Significant bits (LSB's). The output is sent byte by byte. Each byte has 2 bits of payload (128 bits total); resulting in 64 clock cycles in delivering the complete payload.

Figure 3 shows the flow of the Steganography operation.

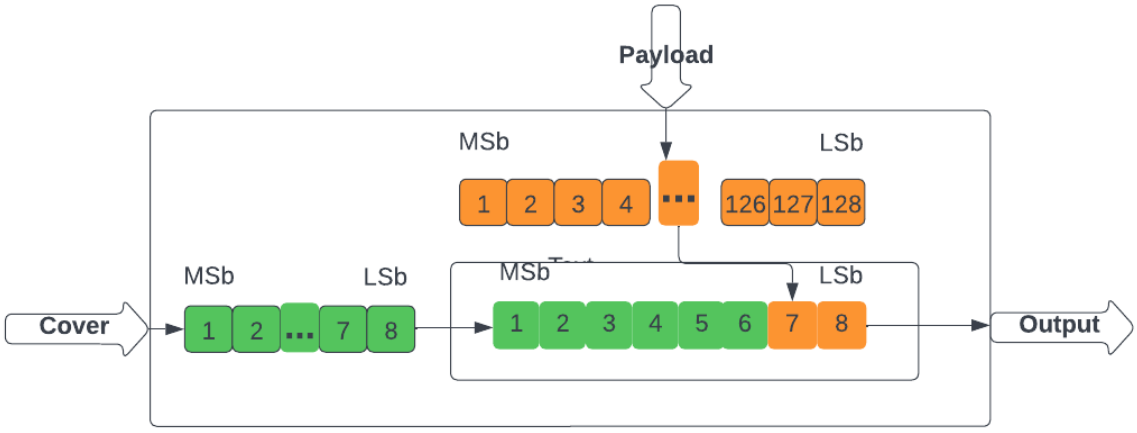

Figure 3: Flow of Steganography module

The simulation results are as shown in Figure 4.

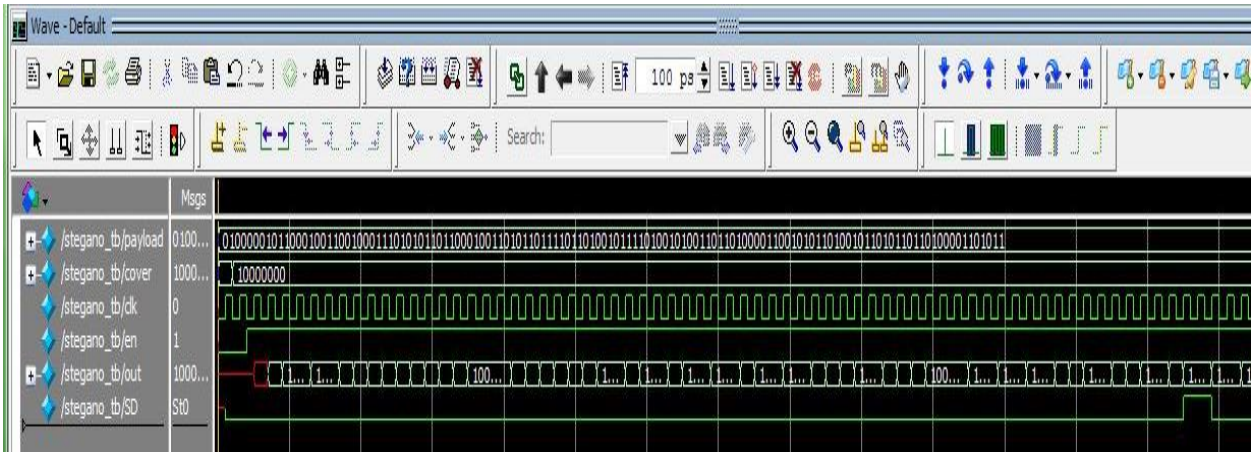

Figure 4: Simulation results of Steganography

3. Module Integration

- The two modules are integrated such that Encryption runs and raises a flag after 23 cycles when it is done. This flag acts as an enable for the steganography block which runs for 65 cycles and raises a flag after embedding all bits of payload (cipher text) in cover data. The steganography flag acts as control for encryption block to transfer the next set of 128 bits of payload to steganography block for next round. Figure 5 shows the flow of the integrated modules.

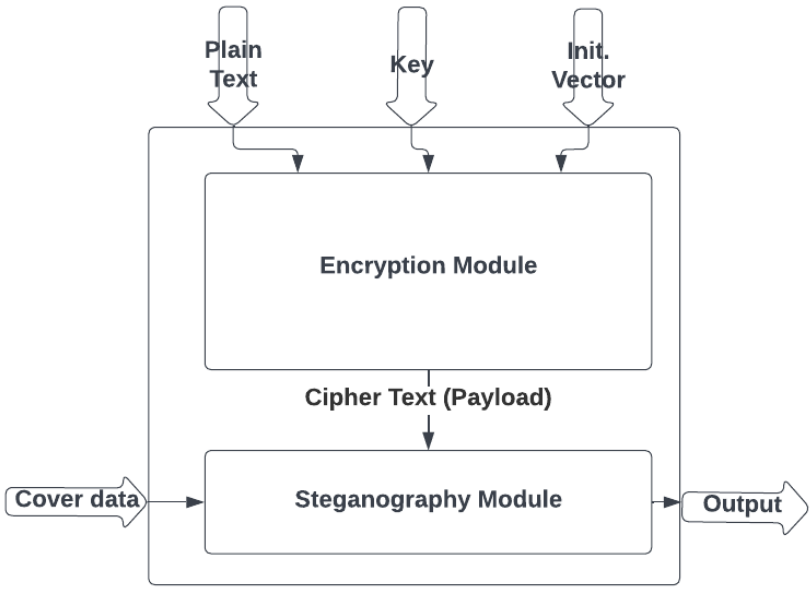

Figure 5: Flow of integrated modules (AES + Steganography)

Figure 6 shows the results after integration of the modules.

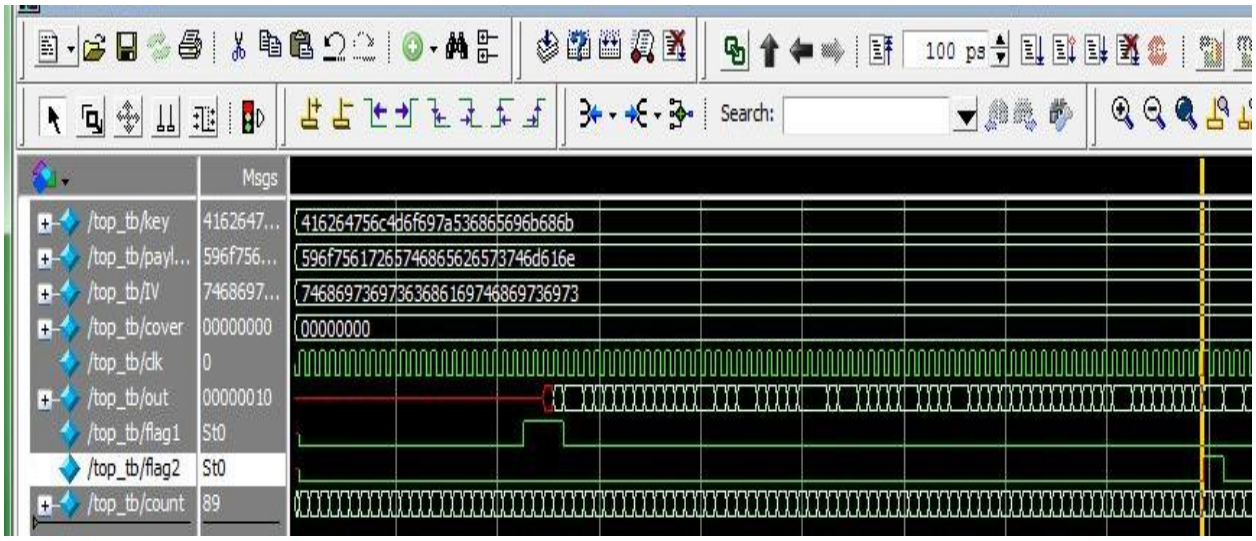

Figure 6: Simulation results of Top integration module

Description of Important Modules

Source files

1. *Aes1281*

   This module takes the input data (state) that has gone through the cbc operation and key (key) as input. The data goes through all steps of AES-encryption by instantiation of modules of expand_key_128 that generates the keys required for the rest of the rounds, one_round that performs one round of AES every two cycles, and final_round that performs the final round of AES and gives the encrypted data (out) as the output.

2. *S_table*

   S_table uses simple binary mathematics to implement the Lookup table instead of using BRAM. It takes the 8 bit input that is to be converted using the lookup table and gives the 8 bit converted output.

3. *Aes1281_cbc*

   This module calls the aes1281 module after performing cbc operation on the input data.

4. *Stegano_core*

   Stegano_core implements the Steganography process. It takes in the encrypted data (payload), cover data (cover) as input and returns the encrypted cum stegnographic coded data as output (out).

5. *Top*

   The top module includes the integration of AES and Steganography and thus calls the aes1281_cbc module as well as the Stegano_core module with the required inputs and outputs.

Simulation files

1. *Aes128_tb*
   Testbench for Aes1281-cbc module.
2. *Stegano_tb*
   Testbench for Stegano_core module.
3. *Top_tb*
   Testbench for top module.

References

[1] Xinmiao Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 9, pp. 957-967, Sept. 2004, DOI: 10.1109/TVLSI.2004.832943.

[2] E. J. Swankoski, R. R. Brooks, V. Narayanan, M. Kandemir and M. J. Irwin, "A parallel architecture for secure FPGA symmetric encryption," 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings., 2004, pp. 132-, DOI: 10.1109/IPDPS.2004.1303101.

[3] P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," 9th EUROMICRO Conference on Digital System Design (DSD'06), 2006, pp. 577-583, DOI: 10.1109/DSD.2006.40.

[4] Y. -H. Chou and S. -L. L. Lu, "A High Performance, Low Energy, Compact Masked 128-Bit AES in 22nm CMOS Technology," 2019 International Symposium on VLSI Design, Automation, and Test (VLSI-DAT), 2019, pp. 1-4, DOI: 10.1109/VLSI-DAT.2019.8741835.

[5] Yiqun Zhang, Kaiyuan Yang, M. Saligane, D. Blaauw, and D. Sylvester, "A compact 446 Gbps/W AES accelerator for mobile SoC and IoT in 40nm," 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), 2016, pp. 1-2, DOI: 10.1109/VLSIC.2016.7573553.

[6] A. Odeh, K. Elleithy and M. Faezipour, "Fast real-time hardware engine for Multipoint Text Steganography," IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014, 2014, pp. 1-5, DOI: 10.1109/LISAT.2014.6845184.

[7] J. Wei, Z. Quan, Y. Hu, J. Liu, H. Zhang, and M. Liu, "Implementing a Low-Complexity Steganography System on FPGA," 2021 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC), 2021, pp. 64-68, DOI: 10.1109/ICWOC52624.2021.9530210.

[8] B. K. Yakti, S. Madenda, S. A. Sudiro, and P. Musa, "Processing Speed Comparison of the Least Significant Bit (LSB) Steganography Algorithm on FPGA and Matlab," 2021 Sixth International Conference on Informatics and Computing (ICIC), 2021, pp. 1-7, DOI: 10.1109/ICIC54025.2021.9632978.