PROJECT REPORT

VLSI IMPLEMENTATION IN HARDWARE SECURITY MODULE BASED ON AES ENCRYPTION METHOD

20ECTE501 - LIVE IN LAB III

Submitted by

AKASH A - 412520106007

JAIGANESH P - 412520106053

MAHIZHAN M - 412520106085

In Partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

SRI SAIRAM ENGINEERING COLLEGE (AUTONOMOUS),

SAI LEO NAGAR, CHENNAI-44

ANNA UNIVERSITY: CHENNAI 600 025

MAY - 2023

SRI SAI RAM ENGINEERING COLLEGE

(An Autonomous Institution; Affiliated to Anna University, Chennai - 600025)

BONAFIDE CERTIFICATE

Certified that this project report titled **"VLSI IMPLEMENTATION IN HARDWARE SECURITY MODULE BASED ON AES ENCRYPTION METHOD"** is the bonafide work of **"AKASH A (412520106007), JAIGANESH P (412520106053) and MAHIZHAN M (412520106085)"** who carried out the 20ECTE501 - LIVE IN LAB III Project Work under my supervision.

SIGNATURE (PROGECT GUIDE)	SIGNATURE (PROGECT LAB INCHARGE)	SIGNATURE (HOD)
Mr.K.SRINIVASAN,MTech	**Ms.S.SARANYA,ME**	**Dr.J.RAJA,M.E,Ph.D.**

.

Submitted for project Viva – Voce Examination held on _ _ _ _ _ _ _ _ _ _ _ _

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Cryptography is very important now-a-days for data security and integrity as the ecommerce and internet applications has increased. But, it has least importance in many cases because of extra memory and other requirements needed for the implementation. The main aim of this work is to implement Advanced Encryption Standard (AES) Encryption using Verilog. To protect data like electronics, cryptographic algorithms are used. Each round of encryption associated with delay can be reduced by AES parallel design. This work proposes a low power and high throughput implementation of AES algorithm using key expansion approach. This minimizes the power consumption and critical path delay using the proposed high-performance architecture. The fundamental goal of the initiative is to increase data flow, although security considerations have become increasingly important over time. The use of encryption and decryption techniques inside VLSI has recently increased since cryptography can convert plaintext to cipher and vice versa. The most recent developments in cryptography technology will be applied in the hardware security module. by simultaneously writing a lot of HDL modules. The main objective is to send and receive data securely without allowing data to be hacked, as well as to improve the performance of a specific parameter. It is interesting to note that any encryption algorithm works in a digital environment and all the blocks in the system will handle digital data in security.

JUSTIFICATION FOR SDG & SAP

SDG No : 9

SAP No : SAP090C

TABLE OF CONTENTS

List of Figures

CHAPTER-1
INTRODUCTION

1.1 OBJECTIVE:

Security is becoming a crucial component of contemporary warfare Cyberattacks account for the majority of attacks. Cyberattack against Systems, Aircraft, Drones, and other Military Equipment, etc.

The initiative's main objective is to increase data flow, but as time goes on, security issues have taken on more significance. Since cryptography can transform plaintext into cypher and vice versa, its use inside VLSI has lately increased. A large number of HDL modules will be simultaneously written in order to implement the most recent advancements in cryptography technology in the hardware security module. The major goal is to send and receive data securely without allowing data to be hacked, as well as to boost the efficiency of a certain parameter. Verilog code was used as the technique in this system. Analog and digital platforms are offered by Xilinx to support the design of both analogue and digital circuits. Interesting fact: Any encryption algorithm will function.

1.2 Motivation:

The primary goal of the project is to increase system security. To that end, we would use our adaptation of AES encryption to strengthen the hardware security module's security in an environment centered around network-centric warfare. Additionally, to strengthen the security of defense technologies from cyber-attacks.

1.3 Relevance of the project:

Security is becoming a crucial component of contemporary warfare. Cyberattacks account for the majority of attacks. Cyberattack against Systems, Aircraft, Drones, and other Military Equipment, etc. These seriously harm a nation's collateral. In order to achieve our goal of improving system security, we would use our adaptation of AES encryption to strengthen the security of the hardware security module, which is an HSM (Hardware Security Module) chip implanted in the system controller.

CHAPTER-2

LITERATURE SURVEY

S.NO	TITLE	AUTHOR	PUBLISHED IN	INFERENCE
1	Journal of Electrical Systems and Information Technology 2 (2015) 178–183	Power efficient and high performance VLSI architecture for AES algorithm K. Kalaiselvi a,∗, H. Mangalam	**2015**	Advanced encryption standard (AES) algorithm has been widely deployed in cryptographic applications. This work proposes a low power and high throughput implementation of AES algorithm using key expansion approach
2	VLSI implementarion of AES Algorithm	Surabh Kumar	**2013**	This paper presents In the past cryptography means only encryption and decryption using secret keys, nowadays it is defined in different mechanisms like asymmetric-key encipherment and

				symmetric-key encipherment
3	VLSI Implementation of Cryptographic Algorithms & Techniques	Favin Fernandes, Gauravi Dungarwal, Aishwariya Gaikwad, Ishan Kareliya, Swati Shilaskar	**2017**	Through the years, the flow of Data and its transmission have increased tremendously and so has the security issues to it. Cryptography in recent years with the advancement of VLSI has led to its implementation of Encryption and Decryption techniques, where the process of translating and converting plaintext into cypher text and vice versa is made possible

CHAPTER-3

EXISTING AND PROPOSED SYSTEM

3.1 EXISTING SYSTEM:

Through the years, the flow of Data and its transmission have increased tremendously and so has the security issues to it. Cryptography in recent years with the advancement of VLSI has led to its implementation of Encryption and Decryption techniques, where the process of translating and converting plaintext into cypher text and vice versa is made possible. In this paper, the review of various aspects of VLSI's implementation of encryption and decryption are covered. Ultimately, with this review, the basic understanding of different VLSI techniques of Encryption and Decryption can be studied and implemented. This is the existing system of the project.

3.2 PROPOSED SYSTEM:

Verilog coding is the method used in this system. We would examine the most recent version of AES encryption first, modify it to reach the algorithm's optimum efficiency, and then implement it in a Hardware Security Module. After that We would implement into a Controller for further study of the Security of the Controller System. This is our proposed methodology of our Project.

CHAPTER-4
REQUIREMENT SPECIFICATION

4.1 Hardware requirements:

4.1.1 Hardware security module:

A hardware security module (HSM) is a physical computing device that safeguards and manages digital keys, performs encryption and decryption functions for digital signatures, strong authentication and other cryptographic functions. These modules traditionally come in the form of a plug-in card or an external device that attaches directly to a computer or network server. A hardware security module contains one or more secure crypto processor chips.

HSMs may have features that provide tamper evidence such as visible signs of tampering or logging and alerting, or tamper resistance which makes tampering difficult without making the HSM inoperable, or tamper responsiveness such as deleting keys upon tamper detection. Each module contains one or more secure crypto processor chips to prevent tampering and bus probing, or a combination of chips in a module that is protected by the tamper evident, tamper resistant, or tamper responsive packaging

Figure 4.1

Uses of HSM:

A hardware security module can be employed in any application that uses digital keys. Typically, the keys would be of high value - meaning there would be a significant, negative impact to the owner of the key if it were compromised.

The functions of an HSM are:

- onboard secure cryptographic key generation
- onboard secure cryptographic key storage, at least for the top level and most sensitive keys, which are often called master keys
- key management
- use of cryptographic and sensitive data material, for example, performing decryption or digital signature functions
- offloading application servers for complete asymmetric and symmetric cryptography.

HSMs are also deployed to manage transparent data encryption keys for databases and keys for storage devices such as disk or tape. HSMs provide both logical and physical protection of these materials, including cryptographic keys, from disclosure, non-authorized use, and potential adversaries. HSMs support both symmetric and asymmetric (public-key) cryptography. For some applications, such as certificate authorities and digital signing, the cryptographic material is asymmetric key pairs (and certificates) used in public-key cryptography. With other applications, such as data encryption or financial payment systems, the cryptographic material consists mainly of symmetric keys.

Figure 4.2

4.1.2 FPGA:

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks. Although one-time programmable (OTP) FPGAs are available, the dominant types are SRAM based which can be reprogrammed as the design evolves.

Applications:

- Aerospace & Defense - Radiation-tolerant FPGAs along with intellectual property for image processing, waveform generation, and partial reconfiguration for SDRs.
- ASIC Prototyping - ASIC prototyping with FPGAs enables fast and accurate SoC system modeling and verification of embedded software
- Automotive - Automotive silicon and IP solutions for gateway and driver assistance systems, comfort, convenience, and in-vehicle infotainment.
- Broadcast & Pro AV - Adapt to changing requirements faster and lengthen product life cycles with Broadcast Targeted Design Platforms and solutions for high-end professional broadcast systems.
- Consumer Electronics - Cost-effective solutions enabling next generation, full-featured consumer applications, such as converged handsets, digital flat panel displays, information appliances, home networking, and residential set top boxes.
- Data Center - Designed for high-bandwidth, low-latency servers, networking, and storage applications to bring higher value into cloud deployments.
- High Performance Computing and Data Storage - Solutions for Network Attached Storage (NAS), Storage Area Network (SAN), servers, and storage appliances.

- Industrial - Xilinx FPGAs and targeted design platforms for Industrial, Scientific and Medical (ISM) enable higher degrees of flexibility, faster time-to-market, and lower overall non-recurring engineering costs (NRE) for a wide range of applications such as industrial imaging and surveillance, industrial automation, and medical imaging equipment.
- Medical - For diagnostic, monitoring, and therapy applications, the Virtex FPGA and Spartan® FPGA families can be used to meet a range of processing, display, and I/O interface requirements.
- Security - Xilinx offers solutions that meet the evolving needs of security applications, from access control to surveillance and safety systems.
- Video & Image Processing - Xilinx FPGAs and targeted design platforms enable higher degrees of flexibility, faster time-to-market, and lower overall non-recurring engineering costs (NRE) for a wide range of video and imaging applications.
- Wired Communications - End-to-end solutions for the Reprogrammable Networking Line card Packet Processing, Framer/MAC, serial backplanes, and more
- Wireless Communications - RF, base band, connectivity, transport and networking solutions for wireless equipment, addressing standards such as WCDMA, HSDPA, WiMAX and others.

Figure 4.3

4.2 Software Requirements:

4.2.1 XILINX ISE:

Xilinx ISE (**I**ntegrated **S**ynthesis **E**nvironment) is a discontinued software tool from Xilinx for synthesis and analysis of HDL designs, which primarily targets development of embedded firmware for Xilinx FPGA and CPLD integrated circuit (IC) product families. It was succeeded by Xilinx Vivado. Use of the last released edition from October 2013 continues for in-system programming of legacy hardware designs containing older FPGAs and CPLDs otherwise orphaned by the replacement design tool, Vivado Design Suite.

ISE enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and ChipScope Pro. The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the ModelSim logic simulator is used for system-level testing.

As commonly practiced in the commercial electronic design automation sector, Xilinx ISE is tightly-coupled to the architecture of Xilinx's own chips (the internals of which are highly proprietary) and cannot be used with FPGA products from other vendors. Given the highly proprietary nature of the Xilinx hardware product lines, it is rarely possible to use open source alternatives to tooling provided directly from Xilinx, although as of 2020, some exploratory attempts are being made.

Simulation:

System-level testing may be performed with ISIM or the ModelSim logic simulator, and such test programs must also be written in HDL languages. Test bench programs may include simulated input signal waveforms, or monitors which observe and verify the outputs of the device under test.

ModelSim or ISIM may be used to perform the following types of simulations:

- Logical verification, to ensure the module produces expected results
- Behavioral verification, to verify logical and timing issues
- Post-place & route simulation, to verify behaviour after placement of the module within the reconfigurable logic of the FPGA

Support:

ISE supports up to Spartan 6, and the older devices including CPLDs (XC9500 and CoolRunner). For development targeting newer Xilinx's devices (7 series, UltraScale and UltraScale+ series), the Xilinx Vivado has to be used.

Xilinx officially supports Microsoft Windows Version 7 64-bit, Red Hat Enterprise 4, 5, & 6 Workstations (32 & 64 bits) and SUSE Linux Enterprise 11 (32 & 64 bits). Certain other Linux distributions can run Xilinx ISE Webpack with some modifications or configurations, including Gentoo Linux, Arch Linux, FreeBSD and Fedora

Figure 4.4

CHAPTER-5

ALGORITHM

5.1 Advanced Encryption Standard:

. The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information.

AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cybersecurity and electronic data protection.

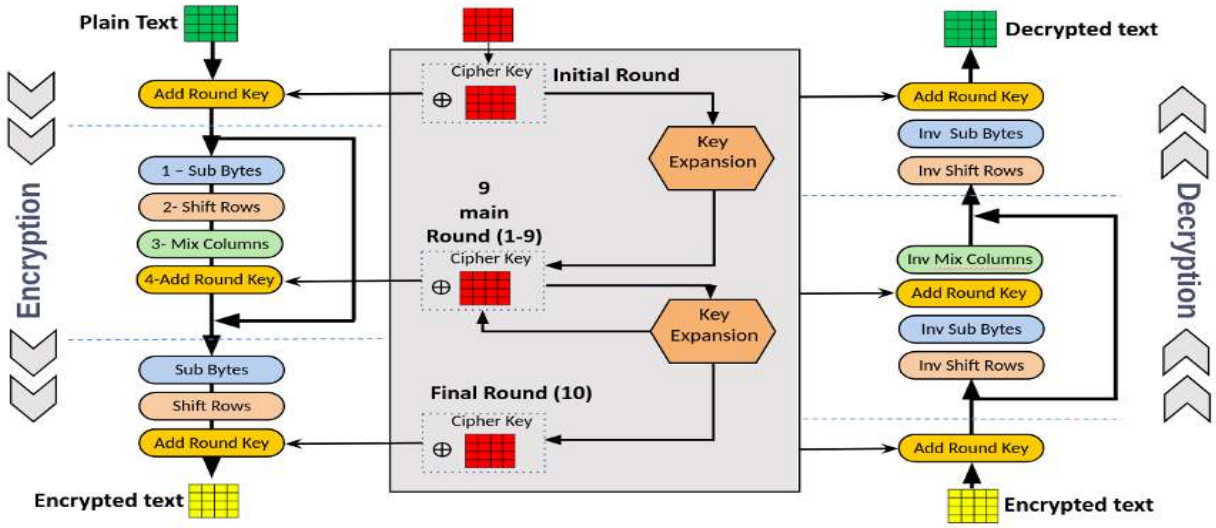

Figure 5.1

Working of AES:

AES includes three block ciphers:

- AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages.
- AES-192 uses a 192-bit key length to encrypt and decrypt a block of messages.
- AES-256 uses a 256-bit key length to encrypt and decrypt a block of messages.

Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively.

Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting. The sender and the receiver must both know -- and use -- the same secret key.

The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of ciphertext.

The AES encryption algorithm goes through multiple rounds of encryption. It can even go through 9, 11, or 13 rounds of this.

Each round involves the same steps below.

- Divide the data into blocks.
- Key expansion.

- Add the round key.
- Substitute/replacement of the bytes.
- Shift the rows.
- Mix the columns.
- Add a round key again.
- Do it all over again.

After the last round, the algorithm will go through one additional round. In this set, the algorithm will do steps 1 to 7 except step 6.

It alters the 6th step because it would not do much at this point. Remember it's already gone through this process multiple times.

So, a repeat of step 6 would be redundant. The amount of processing power it would take to mix the columns again just isn't worth it as it will no longer significantly alter the data.

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array, after which the cipher transformations are repeated over multiple encryption rounds.

The first transformation in the AES encryption cipher is substitution of data using a substitution table. The second transformation shifts data rows. The third mixes columns. The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete

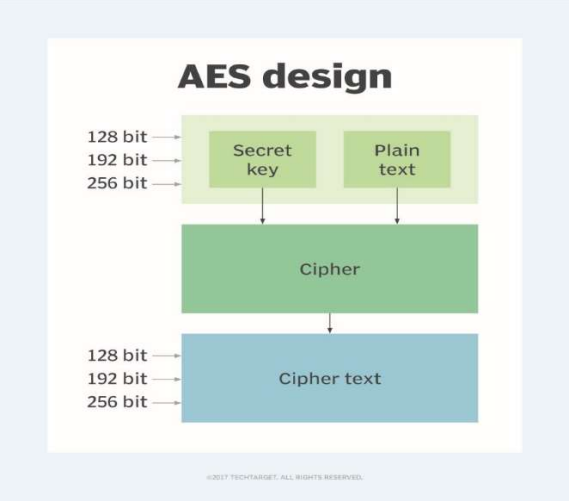

Figure 5.2

Features of AES:

NIST specified the new AES algorithm must be a block cipher capable of handling 128-bit blocks, using keys sized at 128, 192 and 256 bits.

Other criteria for being chosen as the next AES algorithm included the following:

- Security. Competing algorithms were to be judged on their ability to resist attack as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.
- Cost. Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.
- Implementation. Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity.

Difference between AES encryption and DES encryption:

DES served as the linchpin of government cryptography for years until 1999, when researchers broke the algorithm's 56-bit key using a distributed computer system. In 2000, the U.S. government chose to use AES to protect classified information. DES is still used in some instances for backward compatibility.

The two standards are both symmetric block ciphers, but AES is more mathematically efficient. The main benefit of AES lies in its key length options. The time required to crack an encryption algorithm is directly related to the length of the key used to secure the communication -- 128-bit, 192-bit or 256-bit keys. Therefore, AES is exponentially stronger than the 56-bit key of DES. AES encryption is also significantly faster, so it is ideal for applications, firmware and hardware that require low latency or high throughput.

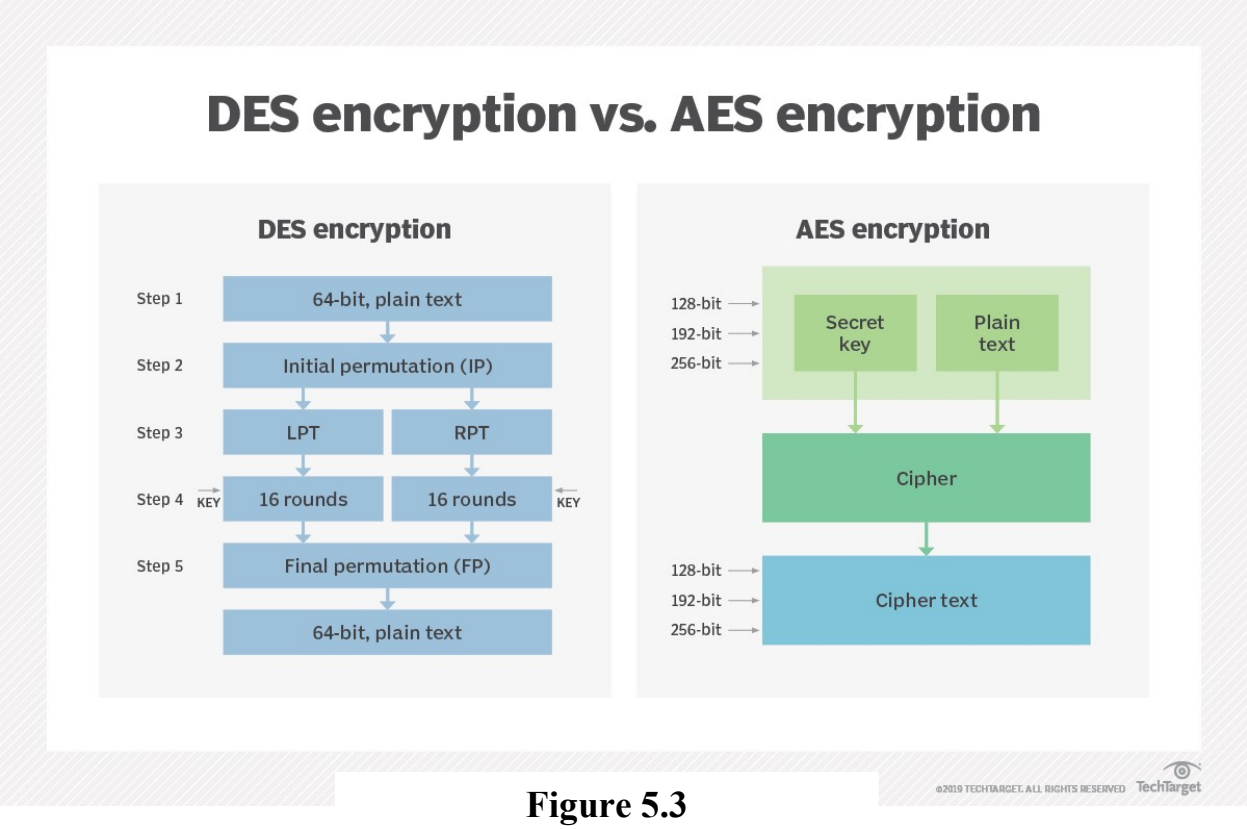

Figure 5.3

AES 256 Encryption:

we know that these encryption algorithms scramble the information it's protecting and turn it into a random mess.

I mean, the basic principle of all encryption *is* each unit of data will be replaced by a different one, depending on the security key.

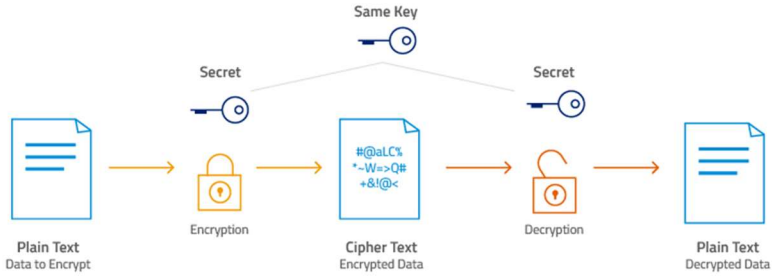

Figure 5.4

There are several rounds in this encryption process,

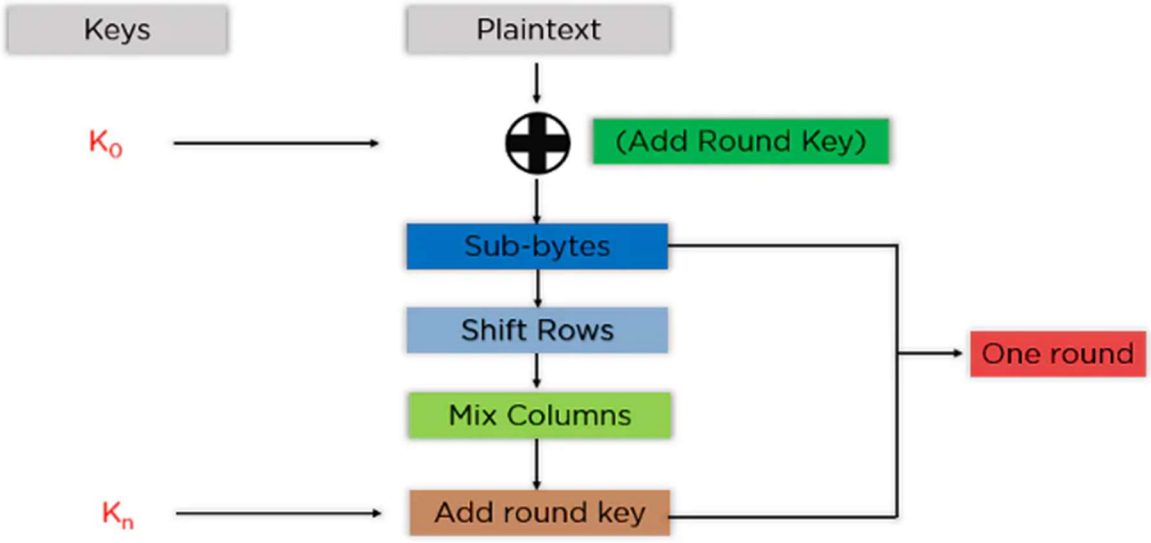

Figure 5.5

Add Round Key: You pass the block data stored in the state array through an XOR function with the first key generated (K0). It passes the resultant state array on as input to the next step.

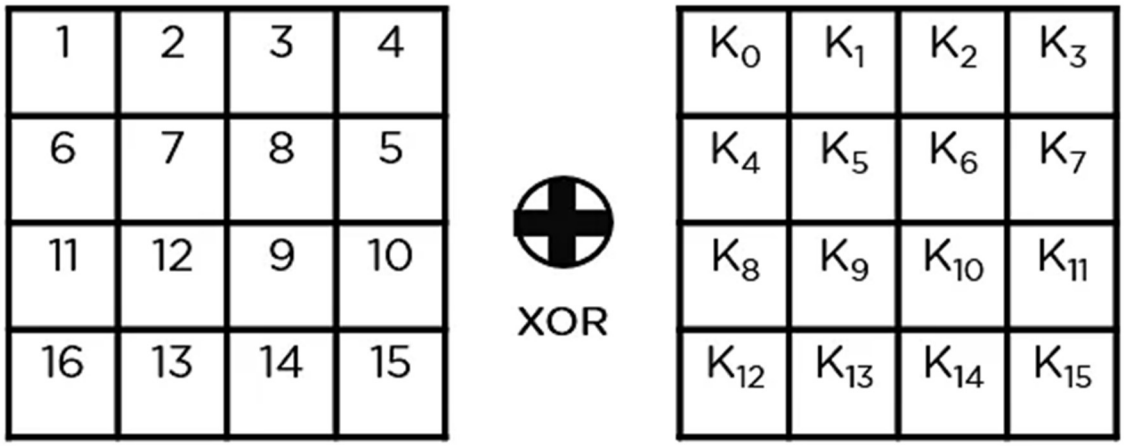

Figure 5.6

Sub-Bytes: In this step, it converts each byte of the state array into hexadecimal, divided into two equal parts. These parts are the rows and columns, mapped with a substitution box (S-Box) to generate new values for the final state array.

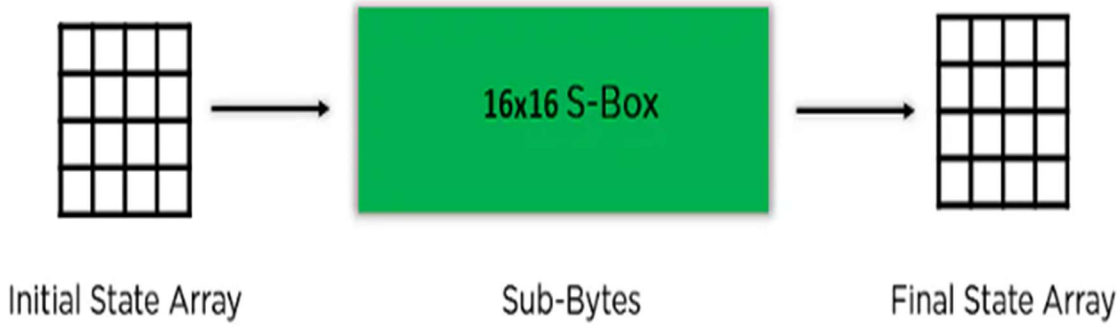

Figure 5.7

Shift Rows: It swaps the row elements among each other. It skips the first row. It shifts the elements in the second row, one position to the left. It also shifts the elements from the third row two consecutive positions to the left, and it shifts the last row three positions to the left.

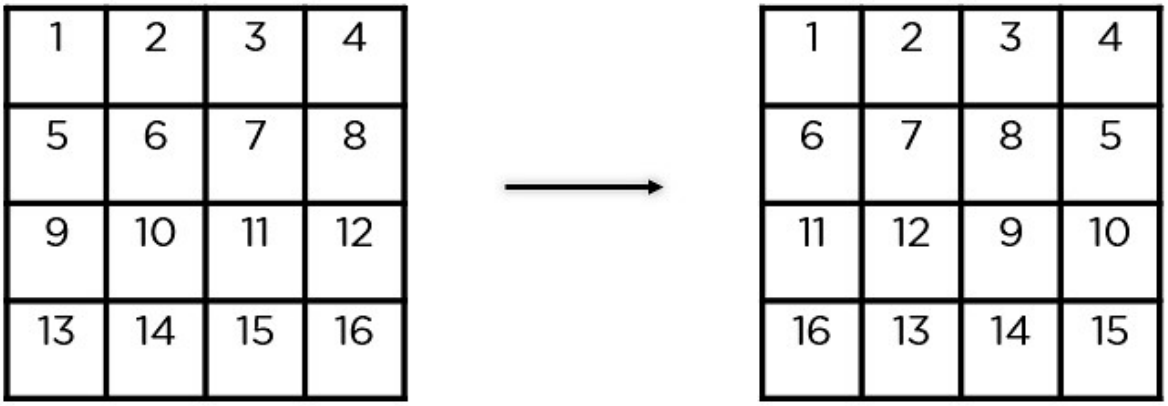

Figure 5.8

Mix Columns: It multiplies a constant matrix with each column in the state array to get a new column for the subsequent state array. Once all the columns are multiplied with the same constant matrix, you get your state array for the next step. This particular step is not to be done in the last round.

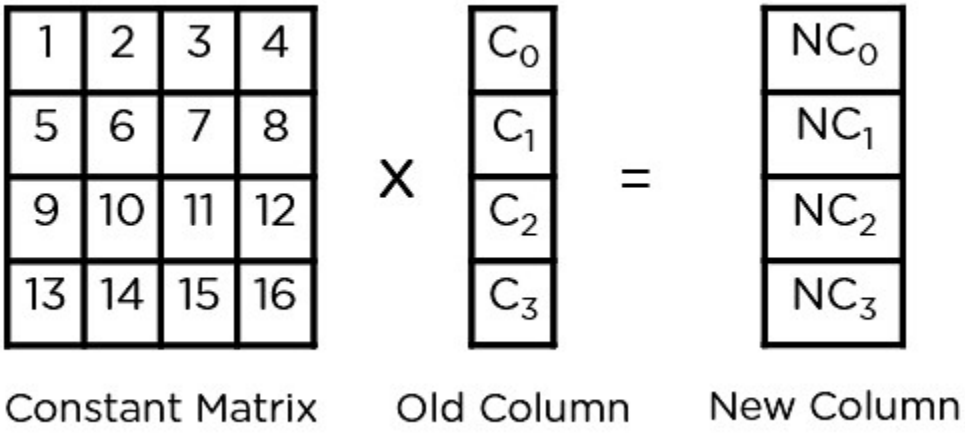

Figure 5.9

Add Round Key: The respective key for the round is XOR'd with the state array is obtained in the previous step. If this is the last round, the resultant state array becomes the ciphertext for the specific block; else, it passes as the new state array input for the next round.

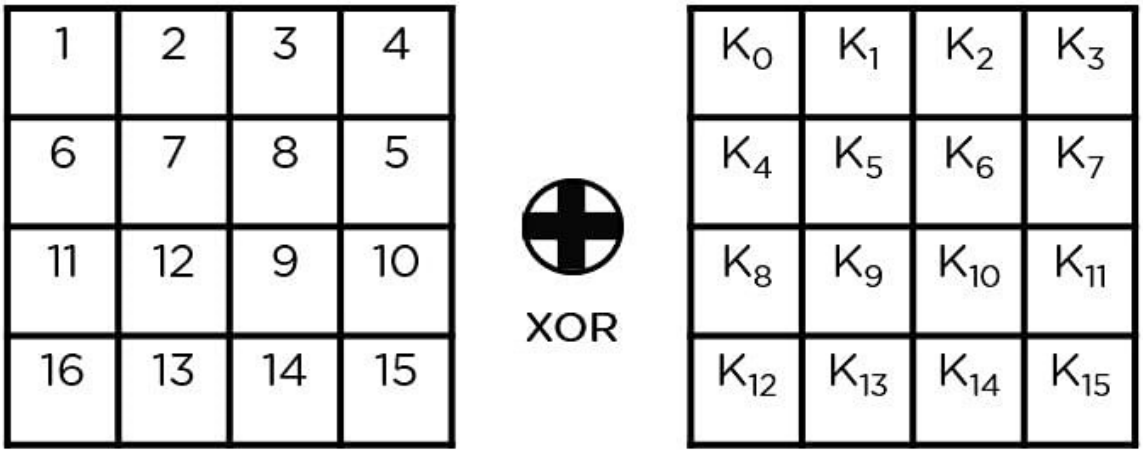

Figure 5.10

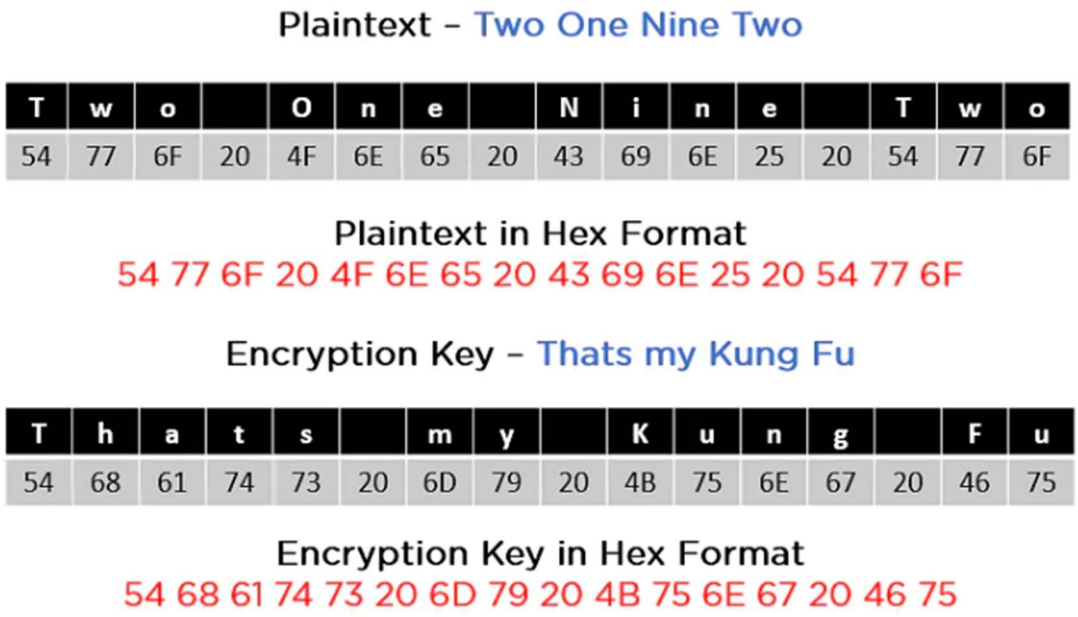

Figure 5.11

As you can see in the image above, the plaintext and encryption convert keys to hex format before the operations begin. Accordingly, you can generate the keys for the next ten rounds, as you can see below

You need to follow the same steps explained above, sequentially extracting the state array and passing it off as input to the next round. The steps are as follows,

Add round key,

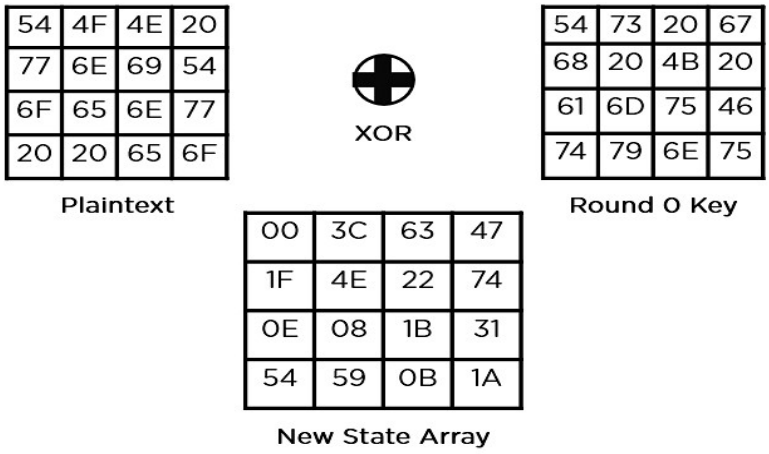

Figure 5.12

Sub-Bytes: It passes the elements through a 16x16 S-Box to get a completely new state array

63	EB	9F	A0
C0	2F	93	92
AB	30	AF	C7
20	CB	2B	A2

New State Array

Figure 5.13

Shift rows,

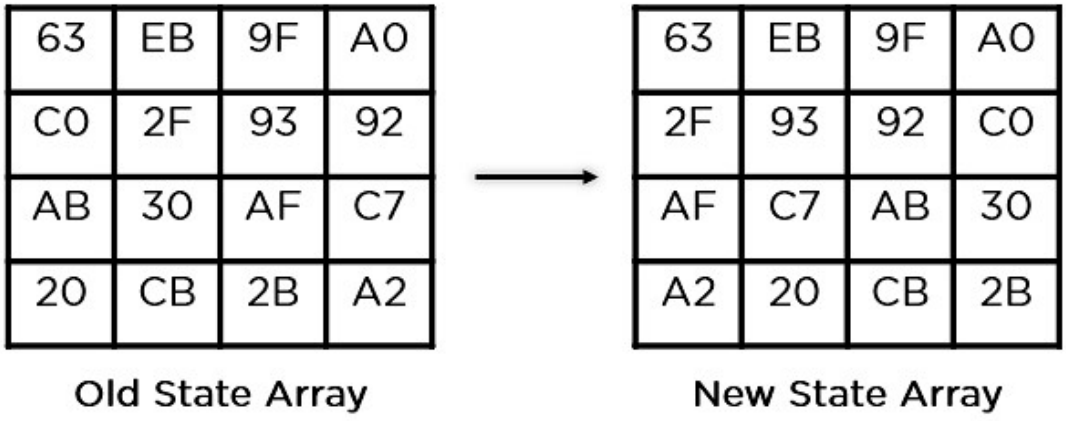

Figure 5.14

Mix columns,

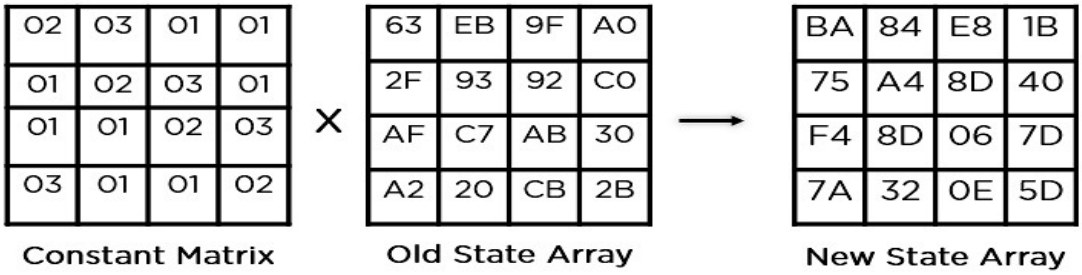

Figure 5.15

Add round key,

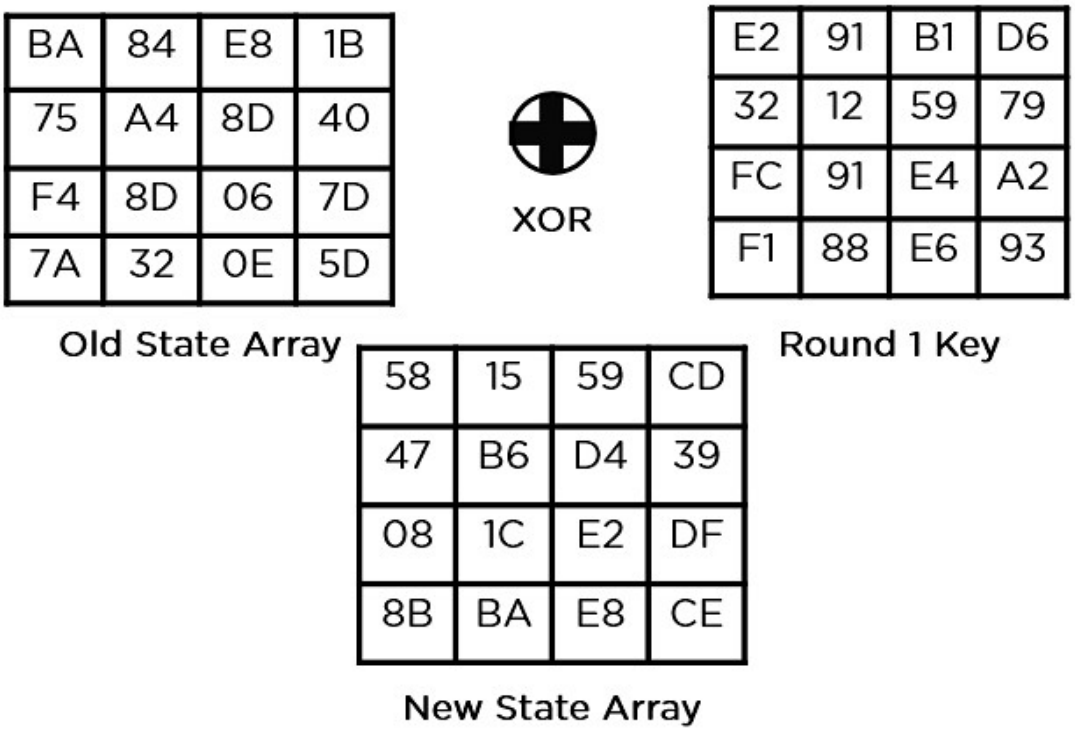

Figure 5.16

This state array is now the final ciphertext for this particular round. This becomes the input for the next round. Depending on the key length, you repeat the above steps until you complete round 10, after which you receive the final ciphertext.

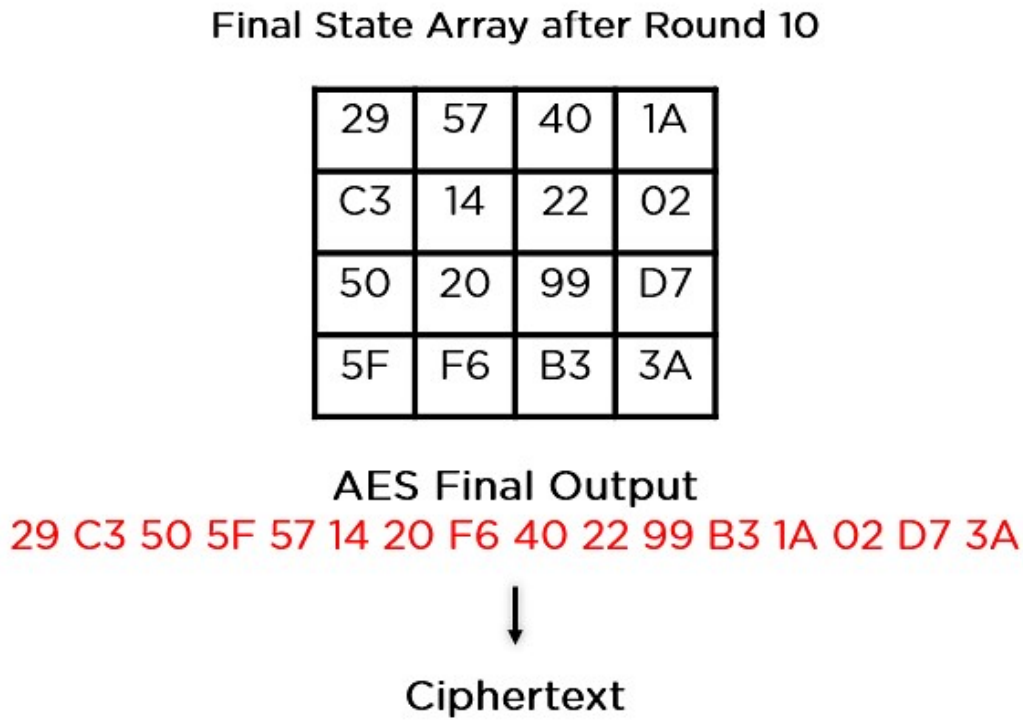

Figure 5.17

Application of AES:

- Wireless Security: Wireless networks are secured using the Advanced Encryption Standard to authenticate routers and clients. Wi-Fi networks have firmware software and complete security systems based on this algorithm and are now in everyday use.

- Encrypted Browsing: AES plays a huge role in securing website server authentication from both client and server end. With both symmetric and asymmetric encryption being used; this algorithm helps in SSL/TLS encryption protocols to always browse with the utmost security and privacy.

- General File Encryption: Apart from corporate necessities, AES is also used to transfer files between associates in an encrypted format. The encrypted information can extend to chat messages, family pictures, legal documents, etc.

- Processor Security: Many processor manufacturers enable hardware-level encryption using the likes of AES encryption to bolster security and prevent meltdown failures, among other low-profile risks

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

This leads us to the conclusion that this project might have research on the definition of cryptography, the AES encryption algorithm, and Xilinx software. From this study, we would modify the AES Encryption algorithm in the best way possible and implement it in the FPGA kit.

6.2 Future scope:

We intended to enhance this project by using an FPGA kit to implement the optimal algorithm in the h circuit. Embed this in a controller so that we could check the controller's security and create a commercial product. the DRDO will receive this for integration into their weapon.

References

1. Kumar, Pramod, T. V. Narendra, and N. A. Vinay. "Short Hand Recognition using Canny Edge Detector." International Journal 7, no. 5 (2017).
2. Kumar, Mamatha MS Pramod, and M.Mamatha. "FPGA Implementation Of Low Area Single Precision Floating Point Multiplier."International Journal of Science Technology and Engineering, Vol.2, no. 2 (2016): 560-566.
3. M.Natheera Banu, FPGA Based Hardware Implementation of Encryption Algorithm, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 8958, Volume-3, Issue-4, April 2014.
4. Deguang Le, Jinyi Chang , Xingdou Gou , Ankang Zhang ,Conglan Lu ,Parallel AES Algorithm for Fast Data Encryption on GPU, IEEE journal on AES 2010.
5. K. Xinmiao Zhang, High speed VLSI architectures for the AES algorithm ,IEEE transactions on VLSI systems, Tech. Rep., sep2004.
6. National Institute of Standards and Technology, Advanced Encryption Standard, Federal Information Processing Standards 197, November 2001.
7. M.Pitchaiah, Philemon Daniel, Praveen, Implementation of Advanced Encryption Standard Algorithm, International Journal of Scientific Engineering Research.