REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE LARBI TEBESSI - TEBESSA
FACULTE DES SCIENCES ET TECHNOLOGIES
DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE

DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE MASTER EN

INSTRUMENTATION

THEME

Description matériel d'un système de cryptage à base d'algorithme AES en langage VHDL

Présenté par :

- BOUTIGHANE El-Houssine

Devant le jury :

- Dr. A. DJARI …………………………...Président

- Dr. H. MAYACHE …………………Encadreur

- Dr. R. SAIDI ....Co-Encadreur

- Dr. T. BENTAHAR ....Examinateur

Année Universitaire 2019 / 2020

PEOPLE'S DEMOCRATIC
REPUBLIC OF ALGERIA

MINISTRY OF HIGHER
EDUCATION AND SCIENTIFIC
RESEARCH

LARBI TEBESSI UNIVERSITY - TEBESSA
FACULTY OF SCIENCES AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER

OF

INSTRUMENTATION

Hardware description of encryption system based on AES algoirthm in VHDL language

Author:

- BOUTIGHANE El-Houssine

Committee :

- Dr. A. DJARI ……………………….….….. Chair

- Dr. H. MAYACHE ……………….……Supervisor

- Dr. R. SAIDI ....Co-Supervisor

- Dr. T. BENTAHAR ....Examiner

Année Universitaire 2019 / 2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الملخــص

يتكون هذا المشروع من تصميم وتنفيذ التشفير وفك تشفير خوارزمية **AES** مع 128 بت من البيانات في دائرة **FPGA.** وقد تم تحسين هذه البنية لتقليل استهلاك الموارد المادية بسرعة تنفيذ أسرع. تمت محاكاة هذه البنية وتوليفها وتنفيذها باستخدام رأس الدائرة 7 **xc7vx330t-3ffg1157** إصدار **Xilinx ISE 14.5..**

الكلمات المفتاحية :
معيار التشفير المتقدم , التشفير , علم فك التشفير , علم التشفير , فك التشفير , التشفير , حقل مصفوفة البوابات المنطقية المبرمجة , كتلة التشفير , رايندال , **addround key, key schedule , shiftrows, subbytes ,mic columns, ,**
inv shiftrows, inv subbytes ,inv mix columns , لغة وصف أجهزة الدوائر المتكاملة عالية السرعة

Abstract

This project consists on the design and the implementation of the encryption and the decryption of AES algorithm with 128 bits of data in FPGA circuit. This architecture was optimized to reduce the consumption of the material resources with a faster speed of execution. This architecture is simulated, synthesized and implemented with the circuit Vertex 7 xc7vx330t-3ffg1157 Edition of XILINX ise 14.5.

Key words: AES, cipher, cryptanalysis, cryptography, decryption, encryption, FPGA, block cipher, Rijndael, addround key, key schedule , shiftrows, subbytes ,mic columns, inv shiftrows, inv subbytes ,inv mix columns ,VHDL .

Résumé

Ce projet consiste en la conception et la mise en œuvre du chiffrement et du déchiffrement de l'algorithme AES avec 128 bits de données en circuit FPGA. Cette architecture a été optimisée pour réduire la consommation des ressources matérielles avec une vitesse d'exécution plus rapide. Cette architecture est simulée, synthétisée et implémentée avec le circuit Vertex 7 xc7vx330t-3ffg1157 Edition de XILINX ise 14.5.

Mots clés: AES, chiffrement, cryptanalyse, cryptographie, déchiffrement, chiffrement, FPGA, chiffrement par blocs, Rijndael, clé addround, calendrier des clés, shiftrows, sous-octets, colonnes de micro, inv shiftrows, sous-octets inv, colonnes de mixage inv, VHDL.

Acknowledgments

Praise be to Allah the Almighty who gives me the strength, the courage and patience to carry out this work.

We would like to thank my Supervisor very much, Dr. H. MAYACHE and my Co-Supervisor Dr. R. SAIDI, for them help, them patience and them valuable advices that has been very helpful to me.

I also thank the members of the jury who honored me by agreeing to judge and enrich this work, and Dr.hamza.ATOUI for his help and encouragement.

my thanks also go to all my teachers for their help and their support.

Dedication

BessmiAllah, this work is the result of many nights, many hard times, failing a lot, but at the end the work is done , with the grease and mercy of ALLAH I was protected , directed , corrected so many times , I am dedicating this thesis to four beloved people who have meant and continue to mean so much to me. Although they are no longer of this world, their memories continue to regulate my life.

First, My father BOUTIGHANE ABD EL FATAH "May God have mercy on him "who has been my source of inspiration especially when I thought of giving up.

Next, My mother for her encouragement and prays of day and night make me able to get such success and honor ".

To my brother, sisters, relatives, mentor, friends who shared their words of advice and encouragement to finish this study.

And To my great teacher Hamza ATOUI , when he change my mind and making me trying to learn programming languages and combine it with my knowledge of electronic to make impact and create a new vision to me as a electronic engineer

Finely, and best friends: Ramzi SGHIR, ALLAG Amer , Said GHERIB , ... who continually provide his moral, spiritual and emotional support.

Table of contents

Chapter I: Introduction to Cryptography

Chapter 02 AES Encryption Algorithm

Chapter 03 : AES Algorithm Implementation

Figures Lists

Figures Lists

Table list

Acronyms list

Term	Definition
AES	Advanced Encryption Standard
P	Plaintext
K	Key
E	Encryption
C	Ciphertext
D	Decryption
SKC	Secret Key Cryptography
PKC	Public Key Cryptography
SSL	Secure Sockets Layer
TLS	Transport Layer Security
GSM	Global System for Mobile Communications
UMTS	Universal Mobile Telecommunications System
EID	Electronic Identity Documents
U.S.A	United States American
NIST	National Institute of Standards associated Technology
FIPS	Federal scientific discipline commonplace
DES	Data Encryption Standard
3DES	Triple Data *E*ncryption Standard
X	Represent Row
Y	Represent Column
K [n]	Bytes with order n of the Key
w[i]	Bytes with the order i in key expansion
IETF	Internet Engineering Task Force
AMD	Advanced Micro Devices

Acronyms list

NET	Network Entity Title
FPGA	Field-programmable Gate Arrays
I/Os	Inputs/Outputs
HDL	Hardware Description Language
VHDL	Very high speed integrated circuits Hardware Description Language
DoD	Department of Defense
IEEE	Institute of Electrical and Electronic Engineers
EDA	Electronic Design Automation
TTM	Time To Market
VHSIC	Very High Speed Integrated Circuits
LUT	Look Up Table
RAM	Random Access Memory
ISE	Integrated Synthesis Environment
CPLD	Complex Programmable Logic Device
UCF	User Constraints File
EDIF	Electronic Data Interchange Format file
PSL	Property Specification Language
MATLAB	Matrix Laboratory
RTL	Register Transfer Level graphical representation

Termonologie

Term	Definition
AES	Advanced Encryption Standard
Array	An enumerated collection of identical entities (e.g., an array of bytes).
Bit	A binary digit having a value of 0 or 1.
Block	Sequence of binary bits that comprise the input, output, State and Round Key. The length of a sequence is the number of bits it contains. Blocks are also interpreted as arrays of bytes.
Byte	A group of eight bits that is treated either as a single entity or as an array of 8 individual bits.
Cipher	Series of transformations that converts plaintext to cipher text using the Cipher Key.
Cipher Key	Secret, cryptographic key that is used by the Key Expansion routine to generate a set of Round Keys
Cipher text	Data output from the Cipher or input to the Inverse Cipher.
Inverse Cipher	Series of transformations that converts cipher text to plaintext using the Cipher Key.
Key Expansion	Routine used to generate a series of Round Keys from the Cipher Key.
Plaintext	Data input to Cipher or output from the Inverse Cipher.
Rijndael	Cryptographic algorithm specified in this Advanced Encryption Standard (AES).
Round Key	Round keys are values derived from the Cipher Key using the Key Expansion routine; they are applied to the State in the Cipher and Inverse Cipher.
State	Intermediate Cipher result that can be pictured as a rectangular array of bytes, having four rows and Nb columns.
Word	A group of 32 bits that is treated either as a single entity or as an array of 4 bytes.

Algorithm Paramerers

Term	Definition
AddRoundKey	Transformation in the Cipher and Inverse Cipher in which a Round Key is added to the State using an XOR operation. The length of a Round Key equals the size of the State (i.e., for Nb = 4, the Round Key length equals 128 bits/16 bytes).
InvMixColumns	Transformation in the Inverse Cipher that is the inverse of MixColumns.
InvShiftRows	Transformation in the Inverse Cipher that is the inverse of ShiftRows.
InvSubBytes	Transformation in the Inverse Cipher that is the inverse of SubBytes
k	Cipher Key.
MixColumns	Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one another) to produce new columns.
Cipher text	Data output from the Cipher or input to the Inverse Cipher.
Nb	Number of columns (32-bit words) comprising the State. For this standard, Nb= 4.
Nk	Number of 32-bit words comprising the Cipher Key. For this standard, Nk = 4.
Nr	Number of rounds, which is a function of Nk and Nb(which is fixed). For this standard, Nr= 10.
Rcon	The round constant word array.
RotWord	Function used in the Key Expansion routine that takes a four-byte word and performs a cyclic permutation.
ShiftRows	Transformation in the Cipher that processes the State by cyclically shifting the last three rows of the State by different offsets.
SubBytes	Transformation in the Cipher that processes the State using a nonlinear byte substitution table (S-box) that operates on each of the State bytes independently.
SubWord	Function used in the Key Expansion routine that takes a four-byte input word and applies an S-box to each of the four bytes to produce an output word.
XOR	Exclusive-OR operation.

General

Introduction

General Introduction

Human beings have always considered information to be a constitutive and determining element in all fields. Since the invention of writing, humanity has expressed the need to transmit their information securely by making it unintelligible to anyone foreign to the exchange, so that the messages cannot be understood, even if they are intercepted.

Therefore, they try to use tools to keep their secrets out of prying eyes: intelligible signs and symbols, figures or colors, use of expressions or sentences agreed to have a specific meaning that differs from the ordinary, etc.

The evolution of these primitive tools through time has made it possible to design more effective and more logical security rules, which gave birth to cryptology. Cryptology is a mathematical science that studies secret communications. It is composed of two complementary fields of study: cryptography and cryptanalysis, the role of cryptographers is to build encryption systems, the objective of cryptanalyses is to "break" these systems.

Cryptography offers a set of techniques to ensure the confidentiality, authentication, data integrity and non-repudiation of the data source. We distinguish two main categories of cryptographic techniques: those with symmetric encryption or with secret keys and those with asymmetric encryption or with public keys. There are several symmetrical algorithms among them we find the DES algorithm and AES algorithm.

The Advanced Encryption Standard (AES), or Rijndael (J. Daeme and V. Rijmen), is an encryption standard used to secure information. AES was published by NIST (National Institute of Standards and Technology). AES is a block cipher algorithm that has been considered extensively and is now widely used. AES is a symmetric block cipher which is intended to replace DES as the approved standard for large applications. [1]

Rijnddael block encryption and decryption was designed by Dr. Joan Daemen and Dr. Vincent Rijmen and the name of the algorithm is a combination of the names of its two creators. It takes an input block of a certain size, usually 128 bits, and produces a corresponding output block of the same sizes. Transformation requires a second entry, which is the secret key. It is important to know this secret key [1].

The main objective of our project is the realization of an IP (Intellectual property) in VHDL for the AES symmetric crypto system for an implementation on an FPGA circuit. Such a system will be able to support embedded applications with the acceleration of their execution by decreasing their encryption and decryption times while reducing the gap between these times.

In this work, the encryption and decryption will be performed with a main length of 128 bits. The input block and the secret key will be given for encryption. Then, the encrypted block and the same secret key will be provided at decryption to obtain the clear block at the output. All encryption and decryption transformations will be described in hardware description language using VHDL standard.

Finally, the VHDL code is synthesized on the FPGA circuit: Virtex-7 XC7VX330T-3FFG1157 Edition using the Xilinx ISE Design Suite 14.5 tool, and simulated on ISim simulator.

A three-part methodological approach was applied:

- The first part provides a brief overview on cryptography, the different encryption, decryption methods, a theoretical description of the different elements of the AES algorithm developed by the National Institute of Standards and Technologies NIST.
- The second part presents the development environment, the description of the hardware architecture of our own design of the circuit developed on an FPGA circuit.
- The last part is endowed with experimental results of simulation, synthesis, implementation and practical tests of the circuit developed on the FPGA platform.

CHAPTER

I

1. Introduction

Nowadays, many data circulates on the different global communication networks all over the world. Many of this information is considered confidential and requires and to be secure. In order to protect this type of data, we proceed to an information transforming process to make it unintelligible to anyone other than the recipient.

Plaintext usually means data in its initial form; it can be read and understood without any treatment. To hide the content of clear text we use encryption, which will result in unreadable gibberish called as ciphertext. In order to recover the content of the text in its initial form, from the ciphertext, we carry out the opposite operation called decryption. This concept has defined under in **Figure1.1** [2]

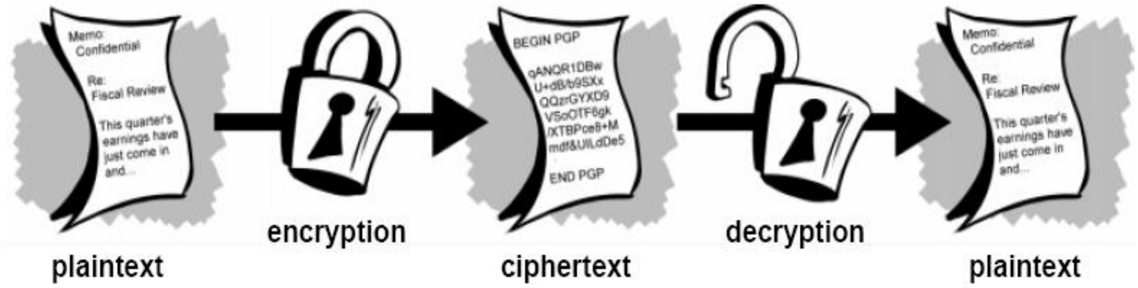


Figure1.1.Encryption and decryption[36]

1.1 What is cryptography?

Cryptology is the science of secrecy, can only really be considered a science since a short time. This science includes cryptography - secret writing - and cryptanalysis - the analysis of the latter. [3]

The word “cryptography” is derived from Greek words that imply mystery writing. Since its inception, cryptography has been used to make information illegible and inaccessible to users without required authorization.

Encrypt information is applying a transformation that modifies the original form of the text which makes it unintelligible for users who do not have the correct encryption key. In its new form, data can be securely sent through transmission channels, or saved, keeping restricted or forbidden access. [4]

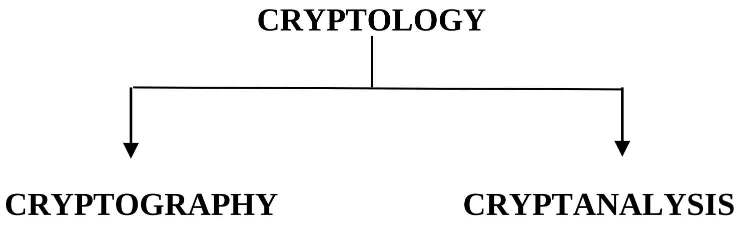


Figure1.2. Cryptology

Cryptography: is the study of mystery composing to conceal the importance of a message.

Cryptanalysis: is artwork of breaking cryptosystems. Maybe you thought that code breaking is for the intelligence community or hackers, and should not be cover in a genuine classification of a scientific discipline. However, most cryptanalysis is carried out by first-rate researchers in academia nowadays. Cryptanalysis is of central significance for cutting-edge cryptosystems: without folks who try to interrupt our crypto methods, we will never know whether they are without a doubt stable or not. [5]

Many years have passed and the cryptography proves always that it is the main center of secretive communications. In its maximum simple form, two people, frequently denoted as Alice and Bob, have agreed on a particular mystery key. At a few later time, Alice may also desire to ship a mystery message to Bob (or Bob would possibly need to ship a message to Alice). The key is used to transform the authentic message (which is typically termed the plaintext) into a scrambled shape that is unintelligible to absolutely everyone who does no longer possess the key. This method is known as encryption and the scrambled message is known as the ciphertext. Whilst Bob receives the ciphertext, he can use the important thing to convert the ciphertext again into the authentic plaintext; this is the decryption process. A cryptosystem constitutes an entire specification of the keys and how they are used to encrypt and decrypt data. [6]

Different types of cryptography's systems of expanding refinement were utilized for some capacities during history. Important applications have included sensitive communications between political leaders and/or royalty, army maneuvers, etc. However, with the development of the internet and applications, which include electronic commerce, many new diverse applications have emerged. These consist of scenarios consisting of encryption of passwords, credit scorecard numbers, email, documents, files, and digital media. [6]

Philosophy and the gigantic range of uses of cryptography must be referenced that cryptographic methodologies are likewise extensively used to protect spared information further to realities this is transmitted starting with one celebration to another. For example, customers may desire to encrypt data stored on laptops, on external difficult disks, inside the cloud, in databases, etc. Additionally, it is probably beneficial to have the ability to carry out computations on encrypted records (without first decrypting the information). The development and deployment of a cryptosystem must address the problem of security. Traditionally, the risk that cryptography addressed became that of an eavesdropping adversary who might intercept the ciphertext and attempt to decrypt it. If the adversary takes place to possess the key, then there is nothing to do. Thus, the main security consideration involves an adversary who does not possess the key, who is still seeking to decrypt the ciphertext. The strategies utilized by the adversary to try to “break” the cryptosystem are termed cryptanalysis. The maximum obvious form of cryptanalysis is to try to wager the key. An attack wherein the adversary attempts to decrypt the ciphertext with every viable key in turn is named an exhaustive key search.

At the point when the enemy try to steal the reasonable key, the plaintext will be found, while some other keys utilized, the "encoded" ciphertext will no doubt be random gibberish. So an apparent first step in designing a steady cryptosystem is to specify a very huge quantity of feasible keys, so many who the adversary will now not be able to check them all in any reasonable amount of time. [6]

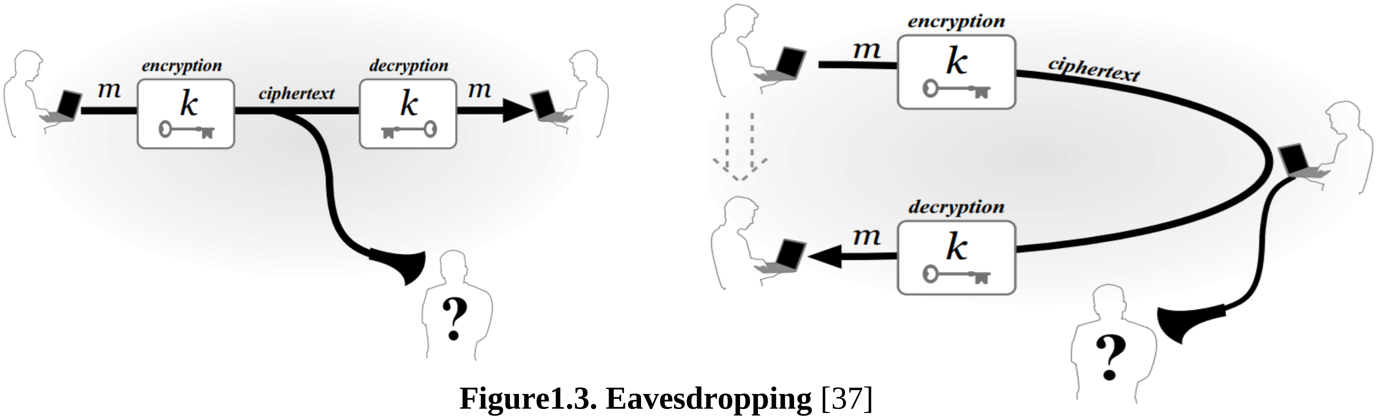


Figure1.3. Eavesdropping [37]

1.2 How does cryptography work?

Ciphering process has a lot of mathematical characteristic used in the coding and decoding operation. A cryptographic algorithm works in combination with some secret records called key that is a phrase, quantity, or word, to encrypt the plaintext. The equal plaintext can be encrypted into distinct ciphertext the use of exceptional keys. Depending on the mechanism used, the key can be used for both encryption and decryption (in that case, the encryption is referred to as secret key or symmetric key encryption), even as different mechanisms use extraordinary keys for the encryption and the decryption procedure (they may be known as public key or asymmetric key encryption). The security of encrypted information is absolutely depending on two parameters: the energy of the cryptographic set of rules and the secrecy of the important thing. [7]

1.3 The purpose of cryptography

Cryptography protects the data that have been transmitted and stored against the unauthorotiesed person plus assuring the arrival of the information to the intended place. In the ideal feel, unauthorized individuals can in no way decrypt an enciphered message. In exercise, reading an enciphered communication can be a characteristic of time; but the attempt and corresponding time this is required for an unauthorized person to decipher an encrypted message can be so huge that it could be impractical. By the point the message is decrypted, the records inside the message may be of minimal value. [8]

There are some particular security requirements, including [9]

• **Authentication**: The technique of proving one's identity. (The primary styles of host-to-host authentication on the Internet these days are name based or address-based, both of which are notoriously weak.)

• **Privacy/confidentiality**: Ensuring that no person can study the message except the supposed receiver.

• **Integrity**: Assuring the receiver that the acquired message has no longer been altered in any manner from the original.

• **Non-repudiation**: A mechanism to prove that the sender sincerely sent this message.

2. Methods of encryption

Cryptography world contain Encryption operation that make data understandable to keep it safer. Encryption makes use of an algorithm known as a cipher and a secret value referred to as the key; if you don't recognize the name of the game key, you can't decrypt, nor can you study any bit of information at the encrypted message—and neither can any attacker. This bankruptcy wills awareness on symmetric encryption, which is the most effective sort of encryption. In symmetric encryption, the important thing used to decrypt is similar to the important thing used to encrypt (in contrast to asymmetric encryption, or public-key encryption, in which the important thing used to decrypt is not like the important thing used to encrypt). You will begin by getting to know approximately the weakest varieties of symmetric encryption, classical ciphers which can be steady against only the maximum illiterate attacker, and then flow directly to the most powerful forms that are secure forever**.** [10]

At the point when a message has scrambling, plaintext refers to the decoded message and ciphertext to the encoded message. A cipher is therefore composed of functions: encryption turns a plaintext right into a ciphertext, and decryption turns a ciphertext lower back into a plaintext. Nevertheless, we will frequently say "cipher" whilst we surely mean "encryption." For example, **Figure1.4**indicates a cipher, E, represented as a box taking as input a plaintext, P, and a key, K, and generating a ciphertext, C, as output. I will write this relation as C = E(K, P). Similarly, whilst the cipher is in decryption mode, I will write D(K, C). [10]

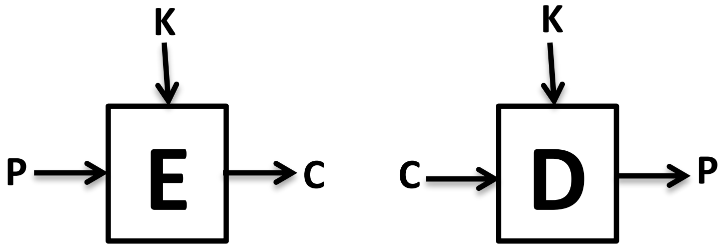


Figure1.4. Basic encryption and decryption [38]

NOTE: For a few ciphers, the ciphertext is the same size as the plaintext; for some others, the ciphertext is slightly longer. However, ciphertexts can never be shorter than plaintexts. [10]

2.1 Symmetric cryptography

When cryptography use the same key for encryption and decryption. Information security depend on the ability of key storing. Compared to its counterpart uneven cryptography, which is explained later, symmetric cryptography may be very rapid and efficient. For this reason, it is miles used extensively for encrypting and decrypting massive files. Even though this system is particularly efficient, it suffers from the subsequent disadvantages:

- **Key Sharing:** Since an initial alternate of the mystery keys required among the parties before they can start encrypting and decryption statistics, secure transmission or sharing of this key will become a problem.

- **Key Management:** A key is required to be shared among every parties who are willing to exchange information securely. Therefore, in a massive network of users who need to alternate facts with others, a unique key is required for every user pair. This storage and control of keys become tough for every consumer who wants to participate is such transactions.

- **Integrity:** Since the receiver cannot verify whether the message has been altered or no longer before receipt, then the integrity of statistics can be compromised.

- **Repudiation:** Since the equal mystery key has to be shared between users, the sender can continually repudiate the messages because there may be no mechanism for the receiver to ensure that the message has been dispatched with the aid of the claimed sender**.** [11]

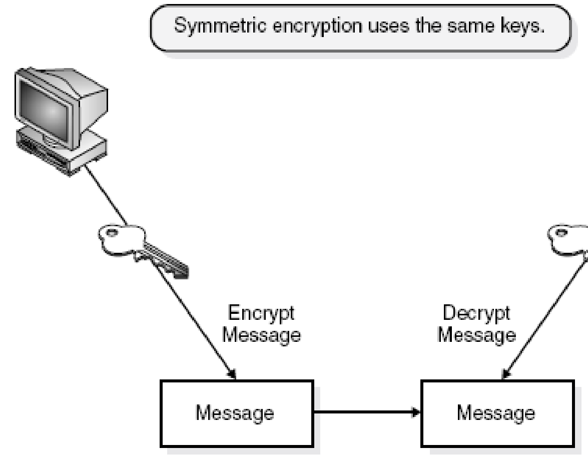


Figure1.5. Symmetric cryptosystem [39]

2.2 Asymmetric cryptography

This method based on public key, coding and decoding information need different keys. These keys named public and individual keys, Public keys are the one, which can be shared, with each person and private keys alternatively are saved secret and known best to the person to whom it belongs. For a celebration to send facts securely to another, they need to encrypt the statistics using the recipient's public key. The recipient can then decrypt these records and recover the message by way of the usage of their corresponding non-public key. Since this personal key is a secret acknowledged handiest to the recipient, the records may be communicated securely without the requirement of initial key trade thus coping with the important thing-sharing problem. Key management also turns into convenient when you consider that there are not any specific keys, which might be required for each user pair inclined to communicate. The person can simply use the recipient's public key and begin secure verbal exchange. Non-repudiation and facts integrity can be finished by using making a party encrypt or sign the message the usage of their private key. This may be verified with the aid of everybody using the signer's corresponding public key. Data can be securely communicated between parties (sender and receiver) alongside with facts integrity and non-repudiation through the subsequent steps:

1. Sender first encrypts the message to be dispatched the use of their non-public key.

2. Sender then encrypts the ensuing ciphertext using the receiver's public key.

The receiver upon receiving the message first decrypts the ciphertext the usage of its private key. This guarantees secure verbal exchange because the receiver's personal key is a mystery acknowledged best to him. The resultant information is then decrypted again using the sender's public key. This offers non-repudiation and statistics integrity since the sender's public secret is known to absolutely everyone and can be used to affirm his identity. [11]

Even despite the fact that asymmetric cryptography offers functions like non-repudiation and records integrity, its execution is still a ways slower than symmetric cryptography making it much less favorable for encrypting and decrypting massive files. [11]

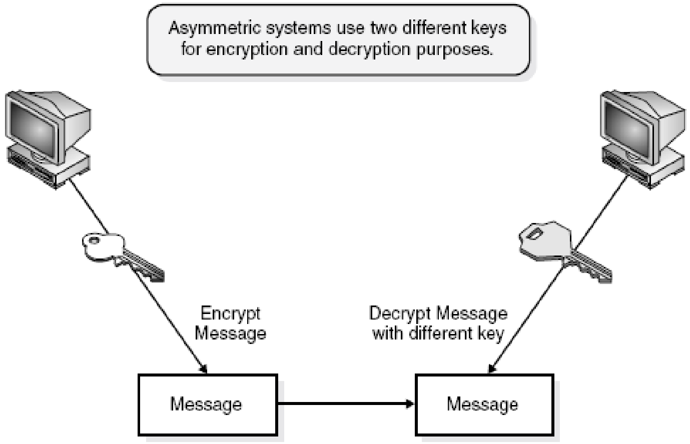
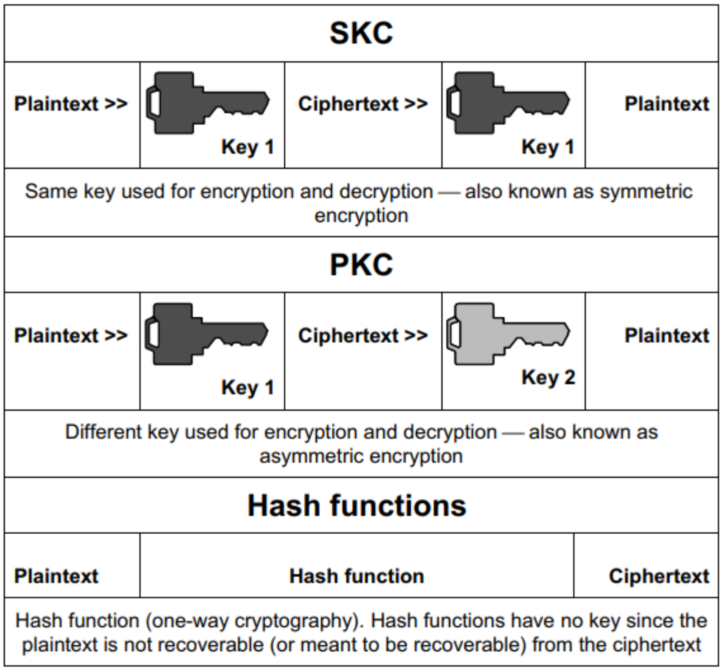

Asymmetric systems use two different keys for encryption and decryption purposes.
Encrypt Message
Decrypt Message with different key
Message
Message

Figure1.6.Asymmetric cryptosystem [39]

3. Types of cryptographic algorithms

Types of Cryptographic Algorithms Three different kinds of cryptographic algorithms. These are[12]:

- Secret Key Cryptography (SKC) — the use of a single key for both encryption and decryption;
- Public Key Cryptography (PKC) — using one key for encryption and any other for decryption
- Hash functions — using a mathematical transformation to irreversibly “encrypt” information.



SKC

Plaintext >> Key 1 Ciphertext >> Key 1 Plaintext

Same key used for encryption and decryption — also known as symmetric encryption

PKC

Plaintext >> Key 1 Ciphertext >> Key 2 Plaintext

Different key used for encryption and decryption — also known as asymmetric encryption

Hash functions

Plaintext Hash function Ciphertext

Hash function (one-way cryptography). Hash functions have no key since the plaintext is not recoverable (or meant to be recoverable) from the ciphertext

Figure 1.7. Three types of cryptographic algorithms [40]

3.1 Secret Key Cryptography (SKC)

SKC, the key used for both encryption and decoding. As demonstrated over, the sender utilizes a key (K1) to encode the plaintext and sends the ciphertext to the recipient. The receiver applies the identical key (K1) to decrypt the message and recover the plaintext. Because a single key is used for each functions, SKC is also referred to as symmetric encryption. In this form of cryptography, the key must be regarded to each the sender and the receiver and this is the secret. The greatest hassle with this shape of cryptography is the distribution of the key. The SKC schemes are typically labeled as being both circulation ciphers and block ciphers. Stream ciphers perform on a single bit at a time and put in force some shape of comments mechanism so that the secret is continuously changing. A block encrypts one block of records at a time using the identical key on every block. [12]

3.2Public Key Cryptography (PKC)

PKC method employs two keys. The first one for encryption , and the second one for the decryption process, using a different keys one for coding and the other to get the clear text proof the effectiveness of PCK .The vital purpose here is that it doesn't matter that secret is applied initial, however that each keys are needed for the method to figure. As a result of a try of keys is needed, this approach is additionally known as uneven cryptography. In PKC, one in every of the keys is selected because the public key and will be publicized as wide as the owner desires. The opposite secret is selected because the personal key and is rarely disclosed to a different party. It is clear-cut to send messages beneath this theme. Suppose Alice desires to send Bob a message. Alice encrypts the message victimization Bob's public key; Bob decrypts the ciphertext using his personal key. This methodology might even be wont to prove who sent a message; Alice, for instance, might code some plaintext along with her personal key; once Bob decrypts victimization Alice's public key, he is aware of that Alice sent the message and Alice cannot deny having sent the message (non-repudiation). [12]

3.3 Hash functions

Hash functions, conjointly alluded to as message digests and unidirectional coding, are calculations that utilization no key. Rather, a fixed-length hash cost is figured dependent on upon the plaintext that makes it unrealistic for either the substance or length of the plaintext to be recuperated. Hash functions are, for the most part, wont to offer an advanced unique mark of a record's substance used to ensure that the document has not been modified by a partner degree intruder or a virus. [12]

It is normally believed that there cannot be two records with a comparative hash cost. This is frequently not right; anyway, the issue is discover two records with a comparative hash cost. It is appallingly problematic to make a record with a comparable hash cost as another document, which is the reason hash vales are utilized in information security and advanced legal sciences. [12]

Hash libraries are sets of hash esteems value notable records. A hash library of notable keen records, suppose, could be an assortment of documents notable to be an area of partner degree OS, though a hash library of notable risky records could be of an assortment of known child attractive pictures utilized in computerized forensics digital crime scene investigation.[12]

3.4 Differences in Encryption Algorithm Types

Since they all perform completely different functions, there are three main sorts of encoding algorithms.

The SKC is ideally suited to encrypting messages. The sender will generate a key for every message to cipher the message. The recipient wants constant key to rewrite the message.

The PKC are often wont to cipher messages however it may be used for nonrepudiation. If the recipient will get the key for the message sent (the session key) encrypted with the sender's non-public key, and then solely this sender may have sent the message.

Hash functions are used for guaranteeing knowledge integrity as a result of any modification created to the contents of a message can lead to the recipient hard a unique hash worth than the one placed within the transmission by the sender. Since it is extremely unlikely that two completely different messages can yield constant hash worth, knowledge integrity is ensured to a high degree of confidence. [12]

4. Application

Discussion about cryptographic tools has finished besides the keys management. We have spent plenteous of this conversation feature issues, which require acting naturally tended to once making decisions with respect to what style of cryptography to actualize and the best approach to help it. It is outstanding that few of those issues were sharp about the applying climate. We tend to in this manner abstained from making any decisions with respect to such issues all through our conversation. On the other hand, maybe, we tend to focused on introducing the executives and cons of significant determinations. [13]

We will currently examine variety of applications of cryptography. Whereas these could be of freelance interest, the important reason we have a tendency to be finding out these applications is to point the kinds of selections taken in specific application environments with reference to problems we left open within the previous chapters. [13]

While these applications are all necessary, and fascinating, their choice has primarily been to produce totally different application environments wherever different selections are taken. We are going to see throughout this discussion that the ‘right’ selections do not seem to be perpetually taken, first time around a minimum of. We are going to conjointly see that a lot of selections are taken supported trade-offs. The chosen applications are:

Cryptography on the Internet: SSL/TLS is one amongst the foremost present cryptographical protocols and provides a superb example of the utilization of hybrid encoding to support open application environments.

Cryptography for wireless local area networks: The development of the cryptography utilized in wireless native space network standards provides variety of necessary lessons in sensible cryptographically style.

Cryptography for mobile telecommunications: GSM and UMTS give smart samples of scientific discipline style in comparatively closed application environments.

Cryptography for secure payment card transactions: The industry has been one amongst the longest business users of cryptography, and a good type of completely different techniques are wont to support differing types of payment transactions.

Cryptography for video broadcasting: Pay-tv provides a desirable application with comparatively simple cryptography requiring the support of fairly subtle key management.

Cryptography for identity cards: The Belgian EID card provides a decent example of a technology creating public-key cryptography wide accessible to be used by different applications.

Cryptography for anonymity: The Belgian EID card provides a decent example of a technology creating public-key cryptography wide accessible to be used by different applications.

Cryptography for digital currency: Bitcoin could be a technology victimization cryptography in a very variety of fascinating ways in which to ascertain a digital currency theme.

It is vital to notice we will not commit to give a comprehensive introduction to those applications, since we have a tendency to be solely fascinated by the role cryptography plays in supporting them. Mainly, for every application, we will be exploring:

- What are the safety requirements?
- What are the appliance constraints that influence decision-making?
- Which cryptanalytic primitives are deployed?
- That cryptanalytic algorithms and key lengths are supported?
- How is vital management conducted?

Once again, we have a tendency to stress that the most reason these specific applications are chosen is illustrative. It is seemingly a number of the cryptanalytic selections taken for these, and similar, applications can amendment over time. [13]

5. Advantage and disadvantage of cryptography

Cryptosystems can be of two types:

- Symmetric Cryptosystems.
- Asymmetric Cryptosystems.

5.1 Symmetric Cryptosystems

A symmetric cryptosystem (or private key cryptosystem) uses just one key for each secret writing and secret writing of the info. The key used for secret writing and secret writing is named the non-public key and solely people that are licensed for the encryption/decryption would are aware of it. In an exceedingly rhombohedral cryptosystem, the encrypted message is shipped over with none public keys hooked up thereto.

5.2 Asymmetric Cryptosystems

In an asymmetric cryptosystem (or public key cryptosystem), there are two completely different keys used for the secret writing and decoding of knowledge. The key used for secret writing is unbroken public then as known as public key, and the decoding secret is unbroken secret and known as non-public key. The keys are generated in such how that it is not possible to derive the non-public key from the general public key. The transmitter and the receiver each have two keys in associate degree uneven system. However, the non-public secret is unbroken private and not sent over with the message to the receiver, though the general public secret is.

CRYPTOSYSTEM	**SYMMETRIC**	**ASYMMETRIC**
ADVANTAGES	• A symmetric cryptosystem is faster. • In Symmetric Cryptosystems, encrypted data can be transferred on the link even if there is a possibility that the data will be intercepted. Since there is no key transmitted with the data, the chances of data being decrypted are null. • A symmetric cryptosystem uses password authentication to prove the receiver's identity. • A system only that possesses the secret key can decrypt a message.	• In asymmetric or public key, cryptography there is no need for exchanging keys, thus eliminating the key distribution problem. • The primary advantage of public-key cryptography is increased security: the private keys do not ever need to be transmitted or revealed to anyone. • Can provide digital signatures that can be repudiated
DISADVANTAGES	• Symmetric cryptosystems have a problem of key transportation. The secret key is to be transmitted to the receiving system before the actual message is to be transmitted. Every means of electronic communication is insecure as it is impossible to guarantee that no one will be able to tap communication channels. Therefore, the only secure way of exchanging keys would be exchanging them personally. • Cannot provide digital signatures that cannot be repudiated	• A disadvantage of using public-key cryptography for encryption is speed: there are popular secret-key encryption methods, which are significantly faster than any currently available public-key encryption method.

Table 1.1Cryptosystem Advantages vs Disadvantage

6. Conclusion

The dissemination of information on a global scale has made it necessary to create common cryptographic systems deemed to be secure. Through this chapter, we were able to browse the science of Information Security. Cryptography is the art of hiding information. The first objective of cryptography is to ensure the confidentiality of data. It responds in particular to the following needs: confidentiality, integrity, authenticity, and signature.

When a cryptographic system is used, some will try to attack it to jeopardize its security. The set of attacks on crypto-systems is cryptanalysis.

Moreover, we explained the different cryptosystem types: symmetric and the asymmetric cryptosystems. The next chapter will allow us to detail in a concrete way the operating mode of the asymmetric cryptosystem by looking more closely at AES encryption algorithm.

CHAPTER

II

1. Introduction

In January 1997, the U.S.A. National Institute of Standards associated Technology (NIST) confirmed the beginning of an advantage to progress a replacement secret writing standard: the AES. The new secret writing usual was to become a Federal scientific discipline commonplace (FIPS), replacement the previous encoding commonplace (DES) and triple-DES. [14]

The National Institute of Standards and Technology set the Advanced Encryption Standard in 2001. The aim is to supply a regular formula for secret writing, sturdy enough to stay U.S. government documents secure for a minimum of subsequent twenty years [2]. Since Rijndael was known because the AES, it is been a hot spot study. Rijndael was allotted all aspects of the study hope to interrupt AES. Years of analysis show that the differential attack remains an efficient thanks to attack block cipher. S box is that the solely nonlinear parts in Rijndael formula, directly poignant their security, therefore rising the S-box against differential attacks performance has been the analysis focus. Associate degree increased Rijndael formula was planned through rising bytes replacement algorithm. The improved formula has stronger ability to resist differential attack, and the avalanche result of the formula is additional affordable. [15]

FIPS (Personal distinguishing proof of Federal laborers and Contractors) - confirmed that principle of cryptography cannot protect the electronic data. The AES rule could be a stellate block cipher which will figure (encipher) and revamp (interpret) information. Cryptography converts data to Associate in nursing unintelligible kind referred to as ciphertext; decrypting the ciphertext converts the information into its original form, referred to as plaintext. [16]

AES can process three different sizes of the key which are : 128 bits, 192 bits, and 256 bits.The corresponding AES algorithms are referred to as AES-128, AES-192, and AES-256, severally. AES supports a set block size: 128 bits. that's to mention, once the secret is determined, AES provides a bijective map from 128-bit plaintext to 128-bit ciphertext, that is, for a key K, AES-128K, AES-192K, AES-256K.[17]

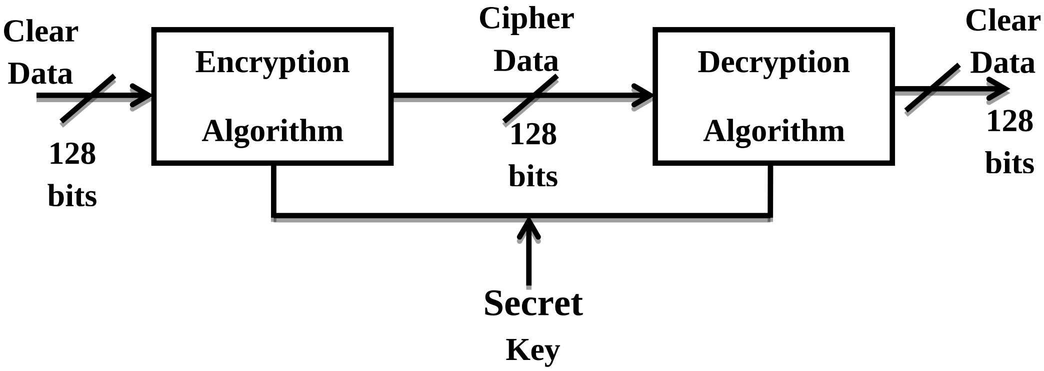


Figure2.1.Overall Representations of Encryption and Decryption

The choice of the AES was a unique jump too far than the old Measurements. During this, the world's best cryptanalysts fixated their consideration on finding even the smallest shortcomings inside the competitor figures submitted to the opposition. when exclusively a few years each up-and-comer algorithmic program was at that point exposed to escalated study, so increasing our confidence within the security of the winning algorithmic program. Of course, the longer the algorithmic program is employed and studied while not being broken; a lot of our confidence can still grow. Today, AES is wide used and no important security weaknesses are discovered. [18]

2. Block cipher

When plaintext blocks, with length n_b, are transformed to ciphertext with the same size using a key also with the same n_b, it means that clear text has coded by block cipher. This set contains a Boolean permutation for every price of the cipher key k. We will tend to solely take into account block ciphers within which the cipher secret is a Boolean vector. If the quantity of bits within the cipher secret is denoted by nk, a block cipher consists of 2nk Boolean permutations. The operation of converting a plaintext block into a ciphertext block is named coding, and the operation of converting a ciphertext block into a plaintext block is named decipherment. Usually, block ciphers are fixed by associate degree coding algorithmic program, being the sequence of transformations to be applied to the plaintext to get the ciphertext. These transformations are operations with a comparatively straightforward description. The ensuing Boolean permutation depends on the cipher key by the very fact that key material, computed from the cipher key, is employed within the transformations.

For a block cipher to be up to its task, it has to fulfil two requirements:

1. Efficiency. Given the worth of the cipher key, applying the corresponding Boolean permutation, or its inverse, is economical, ideally on a large vary of platforms.

2. Security. It should be not possible to use information of the interior structure of the cipher in science attacks. All block ciphers of any significance satisfy these needs by iteratively applying Boolean permutations that are comparatively straightforward to explain. [19]

A block cipher consists of associate encoding rule and a decipherment algorithm:

The encoding rule (E) takes a key, K, and a plaintext block, P, and produces a ciphertext block, C. we tend to write associate encoding operation as C = E(K, P).

The decipherment rule (D) is that the inverse of the encoding algorithm and decrypts a message to the initial plaintext, P. This operation is written as P = D(K, C).

Since they are the inverse of every different one, the encoding and cryptography algorithms sometimes involve similar operations. [20]

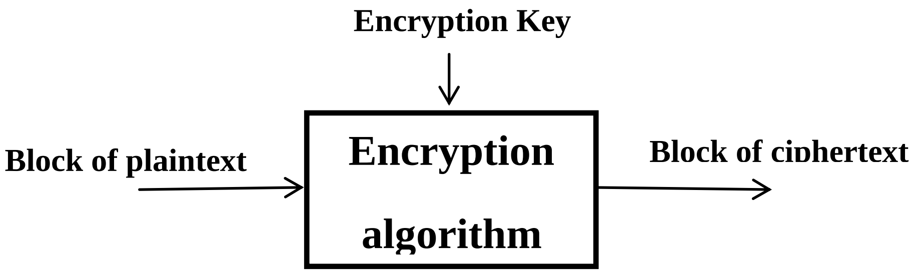


Figure2.2. Model of a block cipher

3. AES algorithm

The U. S. National Institute of commonplaces and Technology (NIST) set block cipher to be alluded to as the advanced coding Standard In Jan 1997, or AES to switch DES. The opposition started with partner degree open includes gatherings to submit applicant block cipher for investigation. [1]

The bilaterally symmetrical coding algorithmic rule (that is currently possibly to be encountered during a new application) is that the advanced coding commonplace or AES. [21]

Joan Daemen and Vincent Rijmen are tow Belgian cryptographers, those names the root of the AES algorithm Rijndael's name. It had been actually a get back of the block cipher and re-intended to manage far-renowned assaults. It had been significantly engaging throughout the AES method as a result of its efficiencies (it is one among the foremost normally economical designs) and therefore the nice scientific discipline properties. Rijndael could be a substitution-permutation network that follows the work of Daemens pH scale.D. Wide-trail style philosophy. It had been tried to resist each linear and differential cryptanalytics (attacks that stony-broke DES) and has excellent applied mathematics properties in alternative regards. In fact, Rijndael was the sole one among the five finalists to be able to prove such claims. The opposite security favorite, Serpent, was conjectured to conjointly resist to identical attacks, however it was less favored as it is much slower. [22]

The way toward picking AES was smart because of partner degree bunch that presented an algorithmic guideline, and was along these lines curious about having its algorithmic standard received, had hearty inspiration to search out assaults on the contrary entries. During this method, the world’s best cryptanalysts targeted their attention on finding even the slightest weaknesses within the candidate ciphers submitted to the competition. When solely many years every candidate algorithmic rule was already subjected to intensive study, so increasing our confidence within the security of the winning algorithmic rule. Of course, the longer the algorithmic rule is employed and studied while not being broken. Lot of our confidence can still grow. Today, AES is wide used and no vital security weaknesses are discovered. [23]

3.1 Specification

Rijndael is an iterated block cipher. The emphases are alluded to as rounds. The amount of rounds, which we will in general mean by Nr, relies on the square length and in this way the key length. In every spherical except the ultimate round, an equivalent spherical operates is applied, when with a special spherical key. The spherical operate of the ultimate round differs slightly. The spherical keys key1, . . . , key Nr are derived from the key k by mistreatment the key schedule formula, A byte, as usual, consists of eight bits, and by a word we tend to mean a sequence of thirty two bits or, equivalently, 4 bytes. Rijndael is byte-oriented. Input and output (plaintext block, key, and ciphertext block) are thought-about as one-dimensional arrays of 8-bit-bytes. Each block length and key length are multiples of thirty-two bits. We tend to denote by Nb the block length in bits divided by thirty-two and by Nk the key length in bits divided by 32. Thus, a Rijndael block consists of Nb words (or four • Nb bytes), and a Rijndael key consists of Nk words (or four • Nk bytes). The subsequent table shows the quantity of rounds Nr in accordance to Nk and Nb: [24]

Nk	Nb				
	4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14

Table 2.1.AES's characteristics Nk and Nb

In explicit, AES with key length 128 bits (and the fastened AES block length of 128 bits) consists of ten rounds.

The spherical operate of Rijndael, and its steps, treat associate degree intermediate result, referred to as the state. The state may be a block of Nb words (or four • Nb bytes). At the start of associate degree cryptography, the variable state is initialized with the plaintext block, and at the top, state contains the ciphertext block.

The intermediate result state is taken into account as a 4-row matrix of bytes with Nb columns. Every column contains one among the Nb words of state.

The following table shows the state matrix within the case of block length 192 bits. We have six state words. Every column of the matrix represents a state word consisting of four bytes. [24]

a0,0	a0,1	a0,2	a0,3	a0,4	a0,5
a1,0	a1,1	a1,2	a1,3	a1,4	a1,5
a2,0	a2,1	a2,2	a2,3	a2,4	a2,5
a3,0	a3,1	a3,2	a3,3	a3,4	a3,5

Table 2.2.AES block length 192 bits

3.2 Description

The AES (Rijndael) blocks figure acknowledges a 128-bits piece plaintext, and produces a 128- bits piece ciphertext underneath the administration of 128-bits, 192-bits, or 256-bits piece mystery key. It's a Substitution-Permutation Network style with one collection of steps known as a circular that are perpetual nine, 11, or multiple times (contingent upon the key length) to outline plaintext to ciphertext. [25]

A single round of AES consists of four steps:

1. SubBytes
2. ShiftRows
3. MixColumns
4. AddRoundKey

Each round uses its own 128-piece round key, that originates from the gave mystery key through a strategy alluded to as a key timetable. Try not to think little of the significance of an appropriately planned key calendar. It distributes the entropy of the key across every of the spherical keys. If that entropy is not unfold properly, it causes every kind of bother resembling equivalent keys, connected keys, and alternative similar characteristic attacks. [25]

AES treats the 128-bit input as a vector of sixteen bytes organized in an exceedingly column major (big endian) 4x4 matrix known as the state. That is, the primary computer memory unit maps to a0,0, the third computer memory unit to a3,0, the fourth computer memory unit to a0,1, and therefore the sixteenth computer memory unit maps to a3,3 [25]

a0,0	a0,1	a0,2	a0,3
a1,0	a1,1	a1,2	a1,3
a2,0	a2,1	a2,2	a2,3
a3,0	a3,1	a3,2	a3,3

Table 2.3The AES State Diagram

The entire forward AES cipher then consists of

```
1. AddRoundKey (round=0)
2. for round = 1 to Nr-1 (9, 11 or 13 depending on the key size) do
        1. SubBytes
        2. ShiftRows
        3. MixColumns
        4. AddRoundKey (round)
```

4. Encryption

Toward the start of the mystery composing or Cipher, the info record and in this manner the information key were followed to the State cluster abuse the shows. at the start the XOR operation ought to be performed between every computer memory unit of the input file and therefore the input key and the output are going to be given because the input of the Round-1. Once associate initial spherical Key addition, the State array is remodeled by implementing a spherical operate ten times, with the ultimate spherical differing slightly from the primary Nr–1 rounds. The ultimate State is then traced to the output. The spherical operate is parameterized employing a key schedule that consists of a one-dimensional array of four-byte words derived mistreatment the Key enlargement routine.[26]

The individual transformations that carried out are listed below.

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

Round	Function
--	Add Round Key(State)
0	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
1	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
2	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
3	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
4	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
5	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
6	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
7	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
8	Add Round Key(Mix Column(Shift Row (Sub Byte(State))))
9	Add Round Key(Shift Row (Sub Byte(State)))

Table 2.4 AES encryption cipher using a 16-byte key

Table 2.4 represents the operation performed at every spherical and its order during which each is allotted. All spherical Nr are identical with the exception of the ultimate round, which does not embrace the MixColumns transformation. Therefore, the cipher text, that is, encrypted knowledge are going to be achieved at the top of the ultimate spherical.

4.1 AES cipher functions

The underneath figure.2.3 shows the most elevated level blocks open inside the AES algorithmic program. Conjointly the fundamental inputs to the system and therefore the outputs from the system were clearly painted. As per the quality, ten spherical for 128 bits key length were allotted within which the last round are performed individually.

For each it’s Cipher and Inverse Cipher, the AES algorithmic program uses a spherical operate that's composed of 4 totally different byte-oriented transformations:

- Byte substitution using a substitution table (S-box)
- Shifting rows of the State array by different offsets
- Mixing the data within each column of the State array
- Adding a Round Key to the State

Previously mentioned works were designated for every individual circular and inside the last around the third capacity, that is, intermixture the information inside each section of the State cluster will not be performed. In this way, the last circular is dispensed separately. Supported the key provided, the new set of keys are generated within the Key growth block and is given to the every spherical at the beginning of the cryptography or Cipher, the input file and therefore the input key were derived to the State array victimization the conventions. At the start the XOR operation ought to be performed between every computer memory unit of the input file and therefore the input key and the output are given because the input of the Round-1. Once AN initial spherical Key addition, the State array is reworked by implementing a spherical operate ten times, with the ultimate spherical differing slightly from the primary Nr–1 rounds. The ultimate State is then derived to the output. The spherical operate is parameterized employing a key schedule that consists of a one-dimensional array of four-byte words derived victimization the Key growth routine.[26]

The individual transformations that carried out are listed below.

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

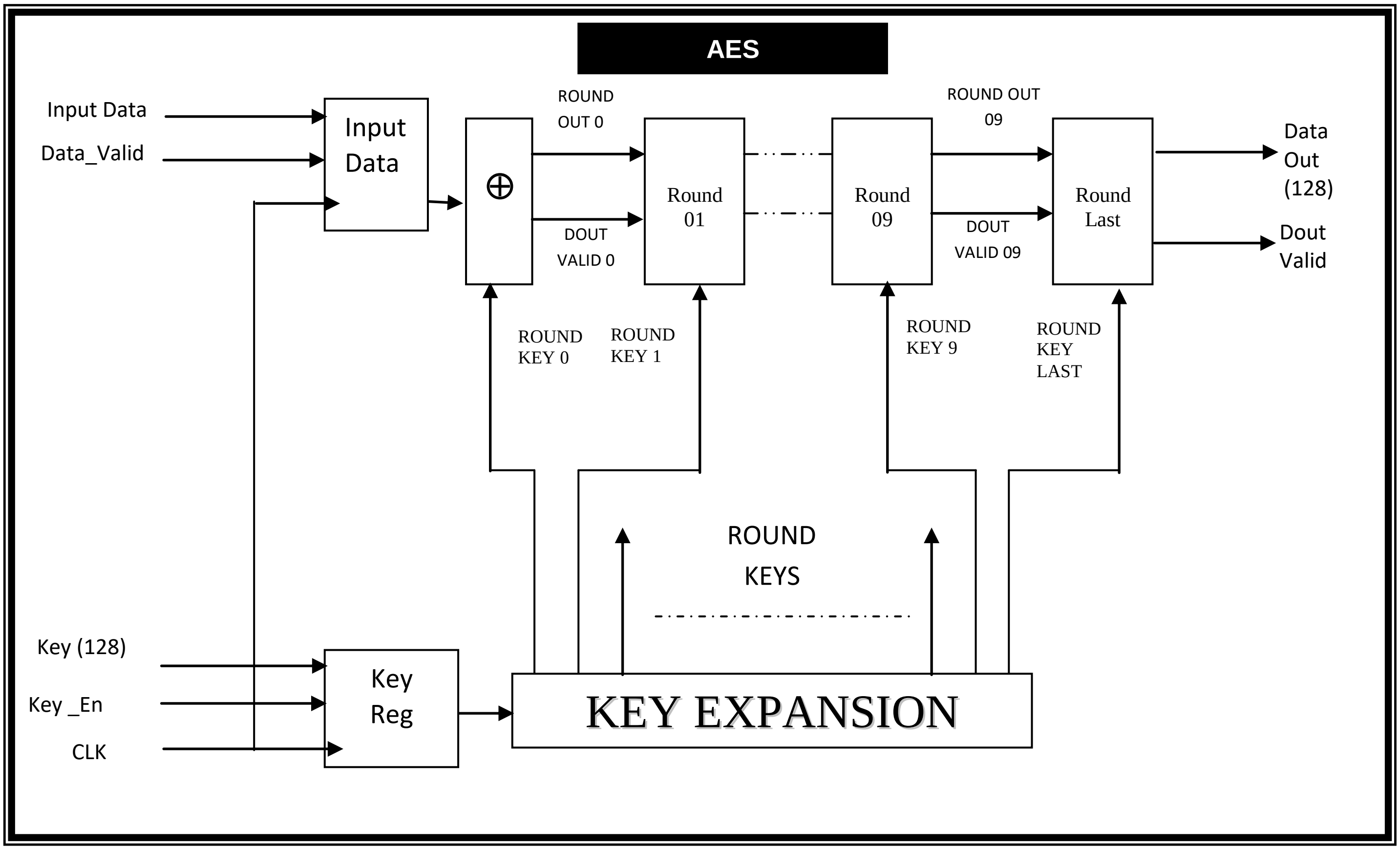


Figure 2.3 Top Level Block Diagram of AES Algorithm [41]

The block diagram shown in the figure 2.3 represents the functions carried out in each round and the functions performed in the last round.

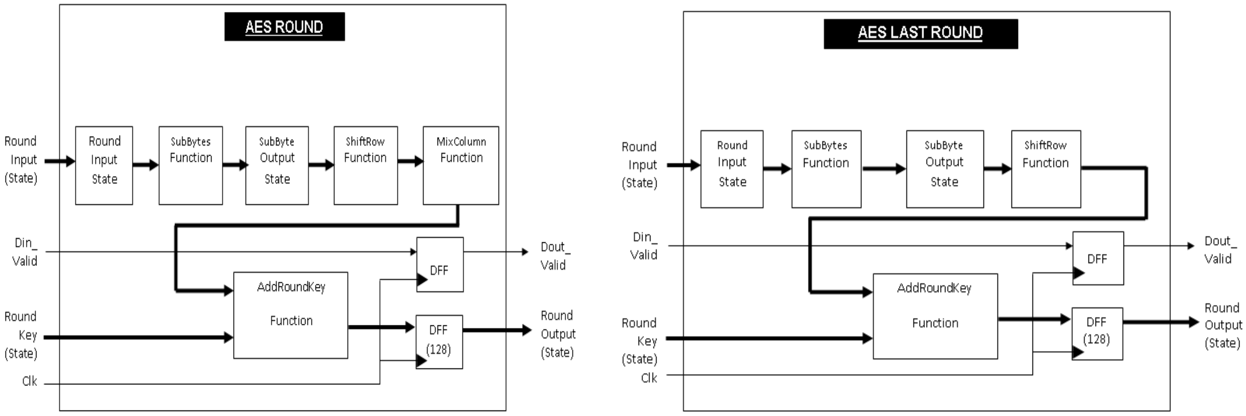


Figure 2.4 Block Diagram for AES Rounds and AES Last Round [41]

4.1.1Addroundkey transformation

AddRoundKey is that the most critical stage in AES rule. Each the key and consequently the information record (additionally commented in light of the fact that the state) are organized in a 4x4 lattice of bytes [30]. What is more, within the stage the subkey is additionally used and combined with state. The most secret is wont to derive the subkey in every spherical by victimisationRijndael's key schedule. The scale of subkey and state is that the same. The subkey is additional by combining every computer memory unit of the state with the corresponding byte of the subkeyvictimisation bitwise XOR [31].

In the AddRoundKey change, a round mystery is extra to the State by a simple bitwise XOR activity. Every of the sixteen bytes of the state is XORed against each of the 16 bytes of some of the enlarged key for the present spherical. The enlarged Key bytes are ne'er reused. Therefore, once the primary sixteen bytes are XORed once more the first 16 bytes of the enlarged key then the expanded key bytes 1-16 are never used again. Ensuing time the Add expanded Key perform is termed bytes 17-32 are XORed against the state. The primary time Add expanded Key is executed.

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	xor	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**	**xor**
Exp Key	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**

Table 2.5.Xor between State and the 1st key expanded

The second time Add Round Key is executed.

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor	xor
Exp Key	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Table 2.6.Xor between State and the 2^{st} key expanded

This process will be continued until the operation ends. The graphical representation of this operation can be seen below.

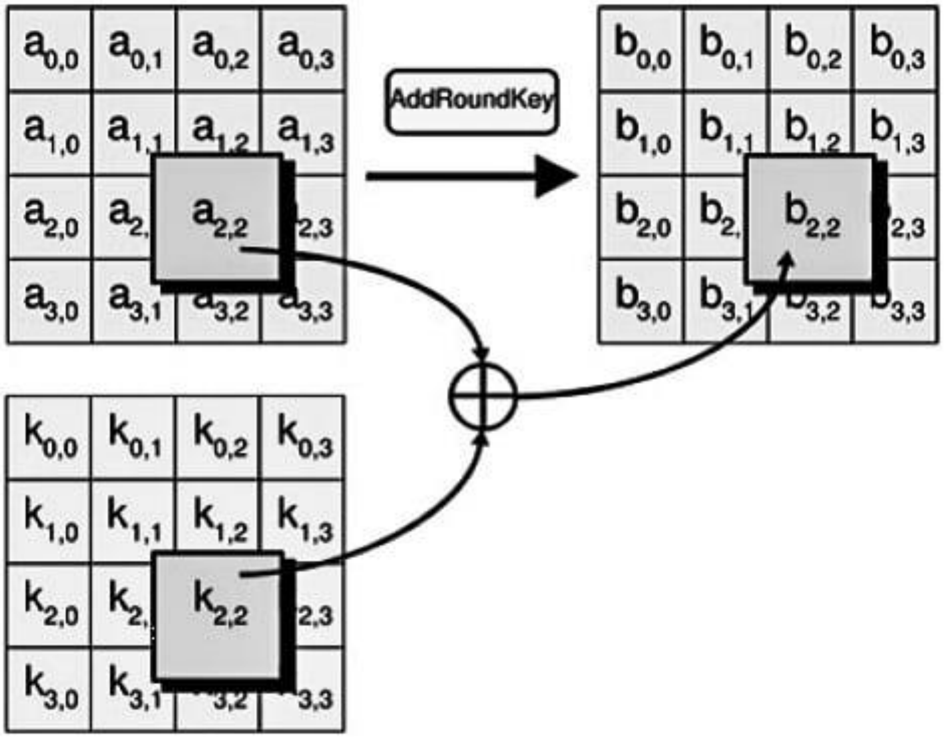


Figure 2.5AddRoundKey Operation [42]

4.1.2 Subbytes transformation

The first stage of every spherical starts with SubBytes transformation. This stage is depends on nonlinear S-box to substitute a computer memory unit within the state to a different byte. Consistent with diffusion and confusion Shannon's principles for cryptographical algorithmic program style its vital roles to get far more security [27]

The SubBytes activity might be a non-direct computer memory unit replacement, in procedure on each computer memory unit of the state severally. The substitution table (S-Box) is invertible and is built by the composition of two transformations:

- Take the multiplicative inverse in Rijndael's finite field
- Apply an affine transformation

Since the S-Box is freelance of any input, pre-calculated forms are used, if enough memory (256 bytes for one S-Box) is on the market. Every computer memory unit of the state is then substituted by the worth within the S-Box whose index corresponds to the value in the state. Figure 2.6 illustrates the result of the SubBytes transformation on the State clearly.

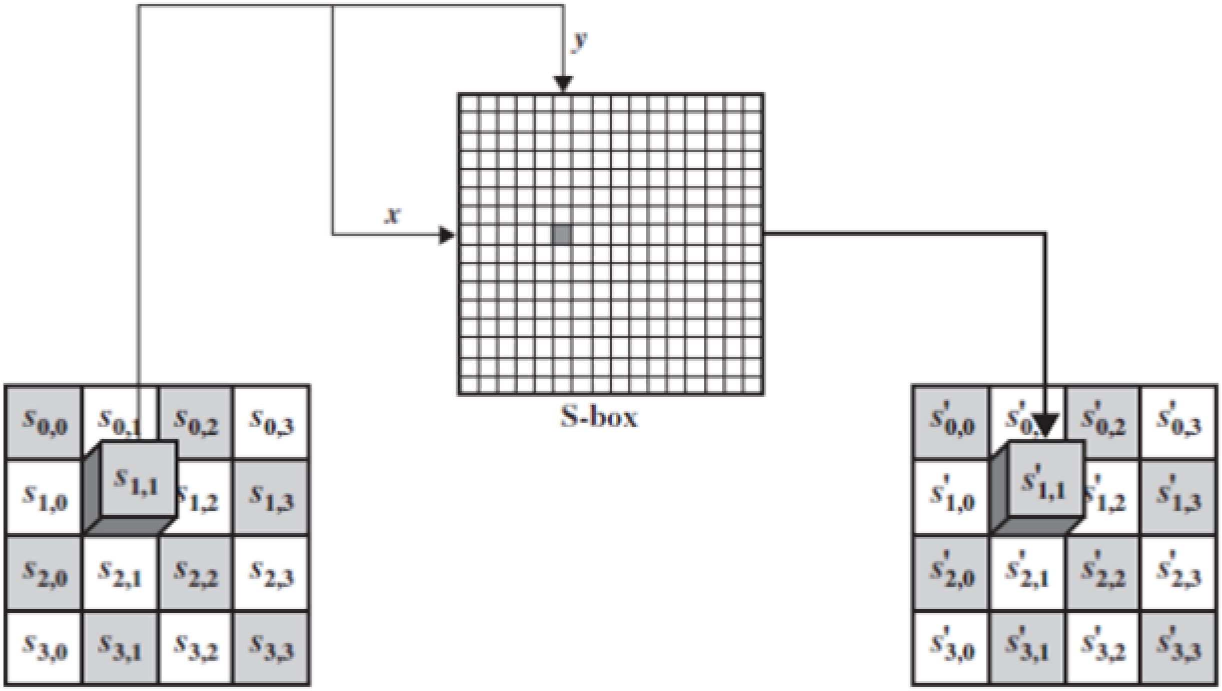
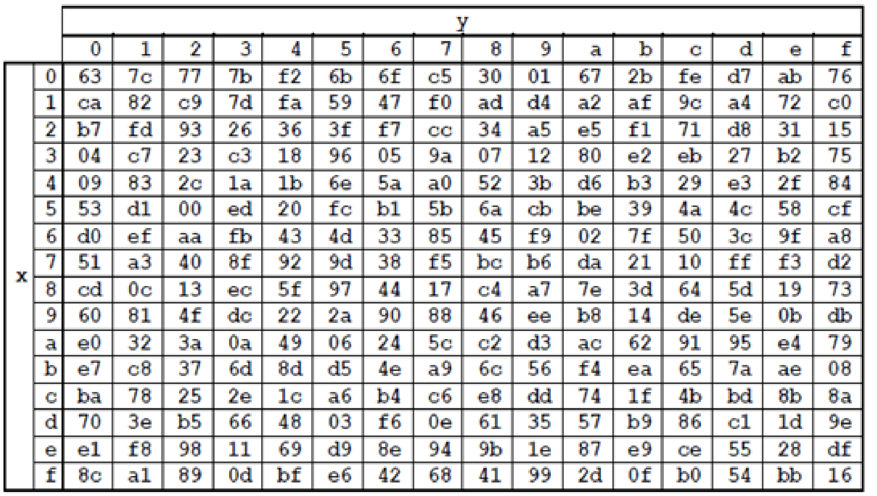

y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Figure 2.6SubBytes Operation of the State [43]

The S-Box for the coding The S-Box are going to be of a 16X16 matrix within which the row is depicted as “x” and therefore the column is represented by “y”. The S-box utilized in the SubBytes transformation is conferred in positional notation type and thus the substitution worth would be determined by the intersection of the row and therefore the column.



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 2.7.S-BOX [44]

4.1.3 Shiftrows transformation

The following stage when SubBytethat perform on the state is ShiftRow. The most arrangement behind this progression is to move bytes of the state consistently to one side in each column rather than line assortment zero. During this technique the bytes of line assortment zero remains and does not play out any stage. Within the initial row, just one computer memory unit is shifted circular to left. The second row is shifted two bytes to the left. The last row is shifted three bytes to the left [28]. The dimensions of recent state is not modified that continues to be because the same original size 16 bytes however shifted the position of the bytes in state as illustrated in Fig 2.4.

Organizes the state during a lattice so plays out a round move for each column. This can be not a limited quantity insightful move. The round move essentially moves each PC memory unit one zone over. A computer memory unit that was within the second position could find yourself in the third position when the shift.

Its round apiece indicates that the PC memory unit inside the last position moved one region could end up in the underlying situation in a similar line. Thence during this ShiftRows operation, every row of the state is cyclically shifted to the left, betting on the row index. This has the impact of moving bytes to “lower” positions within the row, whereas the “lowest” bytes wrap around into the “top” of the row.

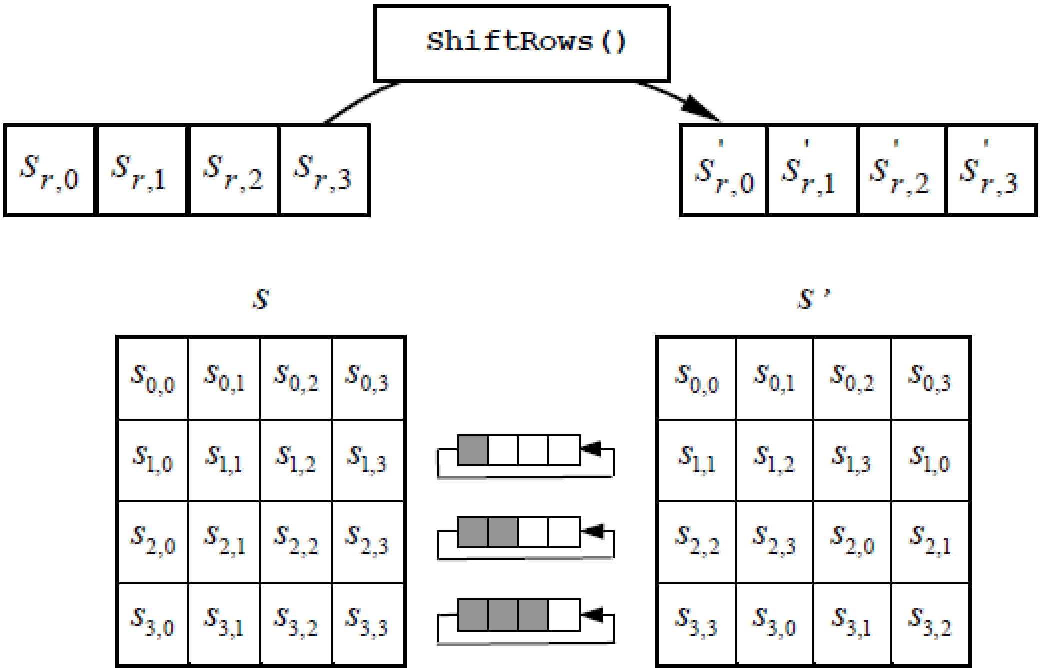

ShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$
s
s'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Figure2.8ShiftRows Operation of the State [45]

Figure 2.8 illustrates the **ShiftRows** transformation. The shifting operation will be carried out horizontally as follows.

- The 1st row is shifted 0 positions to the left.
- The 2nd row is shifted 1 positions to the left.
- The 3rd row is shifted 2 positions to the left.
- The 4th row is shifted 3 positions to the left.

4.1.4 Mixcolumns transformation

Another critical advance occurs of the state is MixColumn. The augmentation is managed of the state. Every computer memory unit of 1 row in matrix transformation multiply by each worth (byte) of the state column. In another word, every row of matrix transformation should multiply by each column of the state. The results of those multiplications are used with XOR to provide a brand new four bytes for subsequent state. During this step the dimensions of state is not modified that remained because the original size 4x4, and so on until all columns of the state are exhausted [29].

In MixColumns activity, components of the state are expanded against those pieces of the lattice. The change works on the State segment by-section. The ingest is organized into a four row table (as represented within the Shift Row function). The multiplication is performed one column at a time (4 bytes). Every worth within the column is eventually increased against each value of the matrix (16 total multiplications). The results of those multiplications are XORed along to provide solely four result bytes for subsequent state. Therefore, four bytes input, sixteen multiplications twelve XORs and four bytes output. The multiplication is performed one matrix row at a time against every worth of a state column.

The pre-defined 4X4 matrix worth and the first column of the ShiftRows state are described as follows, for the multiplication.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

The first result computer memory unit is calculated by multiplying four values of the state column against 4 values of the primary row of the matrix. The results of every multiplication is then XORed to provide one computer memory unit.

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

The second result computer memory unit is calculated by multiplying identical four values of the state column against 4 values of the second row of the matrix. The results of every multiplication is then XORed to provide one computer memory unit.

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

The third result computer memory unit is calculated by multiplying identical four values of the state column against 4 values of the third row of the matrix. The results of every multiplication is then XORed to provide one computer memory unit.

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

The fourth result computer memory unit is calculated by multiplying identical four values of the state column against 4 values of the fourth row of the matrix. The results of every multiplication is then XORed to provide one computer memory unit.

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

This procedure is continual once more with subsequent column of the state, till there are not any a lot of state columns. Therefore, golf shot it all at once; the primary column will embody state bytes 1-4 and can be increased against the matrix within the following manner:

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

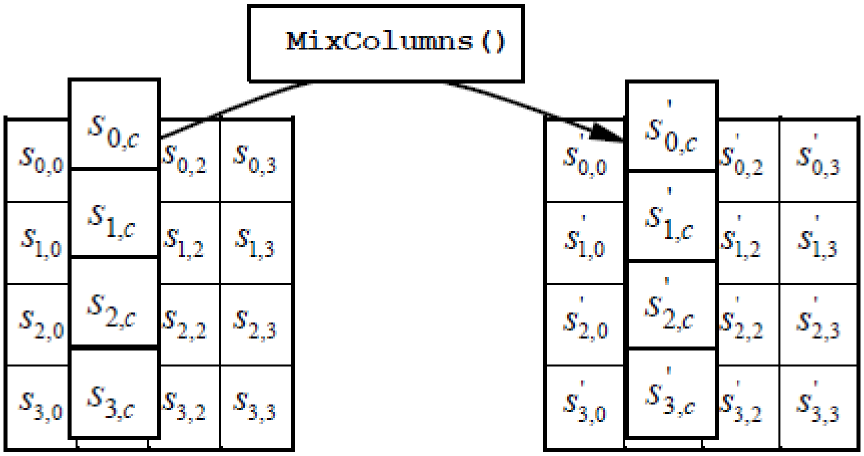

MixColumns()
$s_{0,0}$ $s_{0,c}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,2}$ $s'_{3,3}$

Figure 2.9 MixColumns operates on the State column-by-column[46]

Hence, the picturing of the MixColumns operation depicted higher than offers the clear read on this transformation.

The higher than figure 2.9 represents the clear read on the AddRoundKey transformation that takes place between the results of MixColumns and KeyExpansion and provides the resultant matrix that's used because the input to future spherical.

4.2 Key expansion

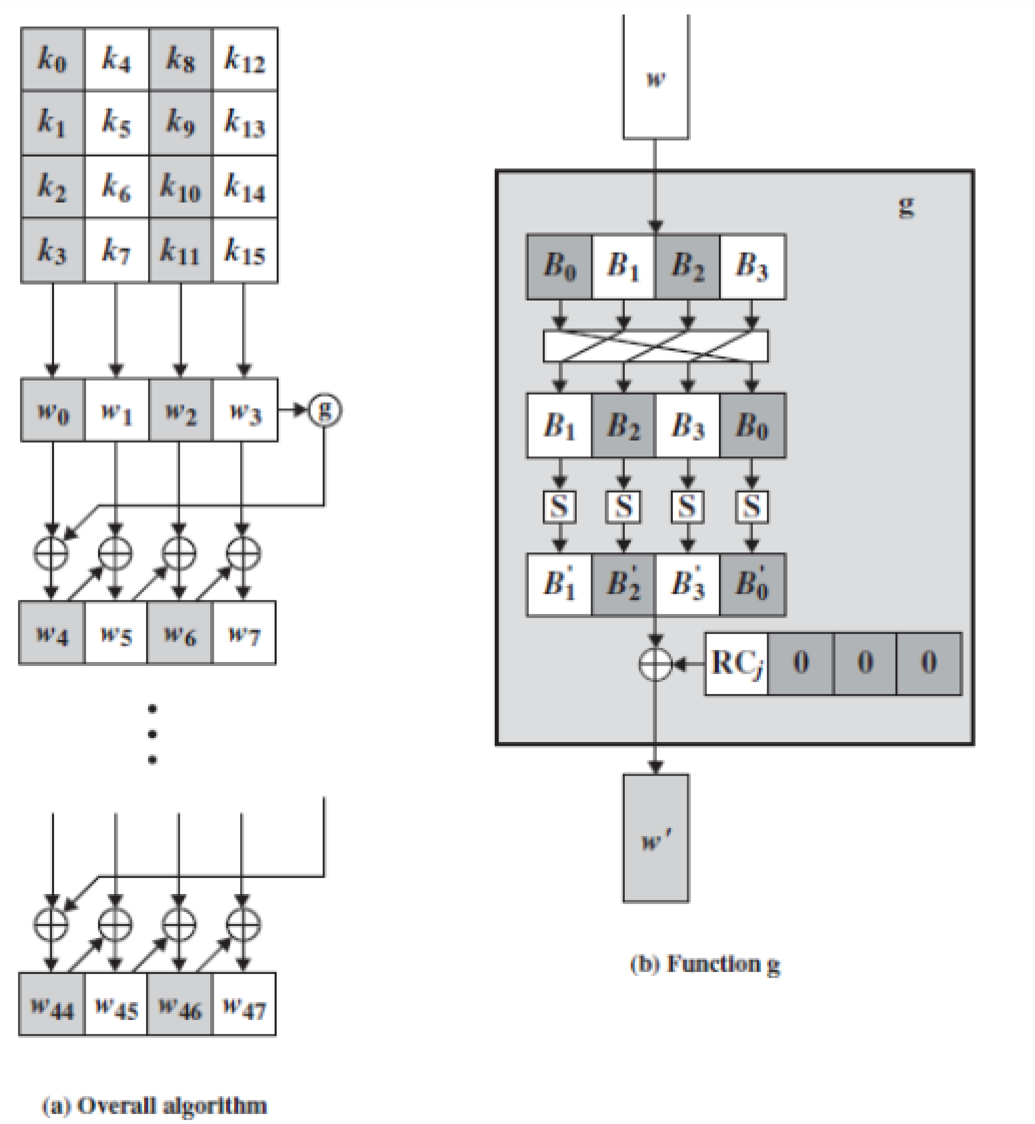
AES algorithmic program depends on AES key development to figure and translate data. It is another most fundamental stride in AES structure. Every spherical encompasses a new key. During this section concentrates on AES Key growth technique. The key growth routine creates spherical keys word by word, wherever a word is an array of 4 bytes. The routine creates 4x (Nr+1) words. Wherever Nr is that the range of rounds [32]. The method is as follows:

The cipher key (initial key) is employed to form the primary four words. The scale of key consists of sixteen bytes (k0 to k15) as shown in Fig.8 that represents in an array. The primary four bytes (k0 to k3) represents as w0, consecutive four bytes (k4 to k7) in 1st column represents as w1, and so on. We are able to use specific equation to calculate and realize keys in every spherical simply as follows:

- K [n]: w[i] = k [n-1]: w[i] XOR k[n]: w[i].

This equation uses to seek out a key for every spherical instead of w0. For w0, we have to use specific equation that is completely different from on top of equation.

- K[n]: w0 = k [n-1]: w0 XOR SubByte (k [n-1]: w3>>8) XOR Rcon [i].



(a) Overall algorithm

(b) Function g

Figure 2.10 Key expansion [47]

Prior to coding or cryptography, the key should be expanded. The expanded secret's utilized in the Add spherical Key perform outlined on top of. When the Add spherical Key perform is named a special a part of the expanded secret's XORed against the state. So as for this to figure the expanded Key should be giant enough in order that it will give key material for each time the Add spherical Key perform is dead. The Add spherical Key perform is mixed up every round further mutually overtime at starting of the algorithm.

The AES formula takes the Cipher Key, K, and performs a Key growth routine to come up with a key schedule. The Key growth generates a complete of Nb (Nr + 1) words: the formula needs Associate in nursing initial set of Nb words, and every of the Nr rounds needs Nb words of key information. The ensuing key schedule consists of a linear array of 4-byte words.

Since the key size is much smaller than the dimensions of the sub keys, the secret is truly “stretched out” to supply enough key house for the formula. Thence associate in nursing 128 bit secret is expanded to 176-computer memory unit key.

There is a relation between the cipher key size, the amount of rounds and also the ExpandedKey size. For Associate in Nursing 128-bit key, there's one initial AddRoundKey operation and there are ten spherical and every round desires a brand new sixteen computer memory unit key, therefor we have a tendency to need 10+1 RoundKeys of sixteen computer memory unit, that equals 176 computer memory unit. Associate in nursing iteration of the on top of steps is named a spherical. The quantity of rounds of the key growth formula depends on the key size.

Key size (bytes)	**Block Size (bytes)**	**Expansion Algorithm (bytes)**	**Expanded Bytes / Round**	**Rounds Key Copy**	**Rounds Key Expansion**	**Expanded Key (bytes)**
16	**16**	**44**	**4**	**4**	**40**	**176**

Table 2.7 Key Expansion

The principal bytes of the extended key are unendingly competent the key. In the event that the mystery is sixteen bytes in length the essential 16 bytes of the extended key will be a comparative in light of the fact that the first key. If the key size is thirty two bytes then the primary 32 bytes of the expanded key are going to be a similar because the original key. Every spherical adds four bytes to the expanded Key. With the exception of the primary spherical, every round conjointly takes the previous rounds four bytes as input operates and returns 4 bytes. The key expansion routine executes a maximum of four consecutive functions. These functions are:

- ROT WORD
- SUB WORD
- RCON
- XOR

Rot Word (4 bytes)

This does a circular shift on 4 bytes similar to the Shift Row Function. The 4-byte word is cyclically shifted 1 byte to the left.

For Example, let’s take a sequence 1,2,3,4 which will be rotated and obtain the result as 2,3,4,1.

Sub Word (4 bytes)

The Key Schedule uses a similar S-Box substitution because the main formula body. This step applies the S-box price substitution as delineated in SubBytes operate to every of the four bytes within the argument. The S-Box is gift within the Appendix-1 for the reference.

Rcon

Basically, this function returns a 4-byte value based on the following table.

Round Number	Rcon	Value
1	Rcon(1)	01000000
2	Rcon(2)	02000000
3	Rcon(3)	04000000
4	Rcon(4)	08000000
5	Rcon(5)	10000000
6	Rcon(6)	20000000
7	Rcon(7)	40000000
8	Rcon(8)	80000000
9	Rcon(9)	1B000000
10	Rcon(10)	36000000

Table 2.8Rcon Table

The results of the SubWords ought to be XORed with the on top of mentioned Rcon values with relation to the corresponding spherical variety. It may be seen that the primary Nk words of the swollen key are crammed with the Cipher Key. Each following word, w[i], is adequate the XOR of the previous word, w [i-1], and therefore the word Nk positions earlier,

w[i-Nk]. For words in positions that are a multiple of Nk, a metamorphosis is applied to w[i-1] before the XOR, followed by an XOR with a spherical constant, Rcon[i].

Steps in Key Expansion

- The first n bytes of the expanded key are simply the cipher key (n = the size of the encryption key)
- The Rcon value i is set to 1
- Until we have enough bytes of expanded key, we do the following to generate n more bytes of expanded key (please note once again that "n" is used here, this varies depending on the key size)
    1. we do the following to generate four bytes
        - we use a temporary 4-byte word called t
        - we assign the previous 4 bytes to t
        - we perform the key schedule core on t, with i as Rcon value
        - we increment i
        - we XOR t with the 4-byte word n bytes before in the expandedKey (where n is once 16 bytes)
    2. we do the following x times to generate the next x*4 bytes of the expandedKey
    3. (x = 3 for n=16)
        - we assign the previous 4-byte word to t
        - we XOR t with the 4-byte word n bytes before in the expandedKey (where n is once 16 bytes)

Hence, for n=16, we generate: 4 + 3*4 bytes = 16 bytes per iteration.

5. DECRYPTION

The decoding is that the technique to get the underlying information that was scrambled. This technique depends on the key that was gotten from the sender of the data. The decoding methods of Associate in Nursing AES is analogous to the coding process within the reverse order and each sender and receiver have constant key to encipher and rewrite knowledge. The last spherical of a decoding stage consists of 3 stages appreciate InvShiftRows, InvSubBytes, and AddRoundKey.

The figure text of 128 bits and furthermore a similar key of 128 bits will be given in light of the fact that the contribution to the disentangling square. The encrypted knowledge are going to be decrypted and also the original plain message will be achieved because the output of the decoding block. The Cipher transformations may be inverted so enforced in reverse order to supply a simple Inverse Cipher for the AES rule. The individual transformations utilized in the Inverse Cipher were listed as follows.

- InvShiftRows
- InvSubBytes
- InvMixColumns
- AddRoundKey

Here additionally ten spherical will be administrated and furthermore the exclusively qualification inside the decipherment hinder with connection to the algorithmic guideline stream is that the aftereffects of the KeyExpansion of each round also will be to the MixCoulmns operation when that the AddRoundKey transformation ought to be carried out.

InvMixColumns (state XOR spherical Key) = InvMixColumns (state) XOR InvMixColumns (Round Key)

The higher than equation represents the fundamental distinction within the method of the AES secret writing and decipherment algorithmic rule

The figure text of 128 bits and furthermore a similar key of 128 bits will be given because the contribution to the decipherment square. The encrypted knowledge are going to be decrypted and also the original plain message will be achieved because the output of the decipherment block. The Cipher transformations are often inverted then enforced in reverse order to provide an easy Inverse Cipher for the AES algorithmic rule.

The individual transformations utilized in the Inverse Cipher were listed as follows.

- InvSubBytes
- InvShiftRows
- InvMixColumns
- AddRoundKey

Here also 10 rounds are distributed and also the solely distinction within the cryptography block with relevancy the algorithmic rule flow is that the results of the KeyExpansion of every round will tend to the MixCoulmns operation when that the AddRoundKey transformation ought to be carried out.

InvMixColumns (state XOR Round Key) = InvMixColumns (state) XOR InvMixColumns (Round Key)

The above equation represents the basic difference in the process of the AES Encryption and Decryption algorithm.

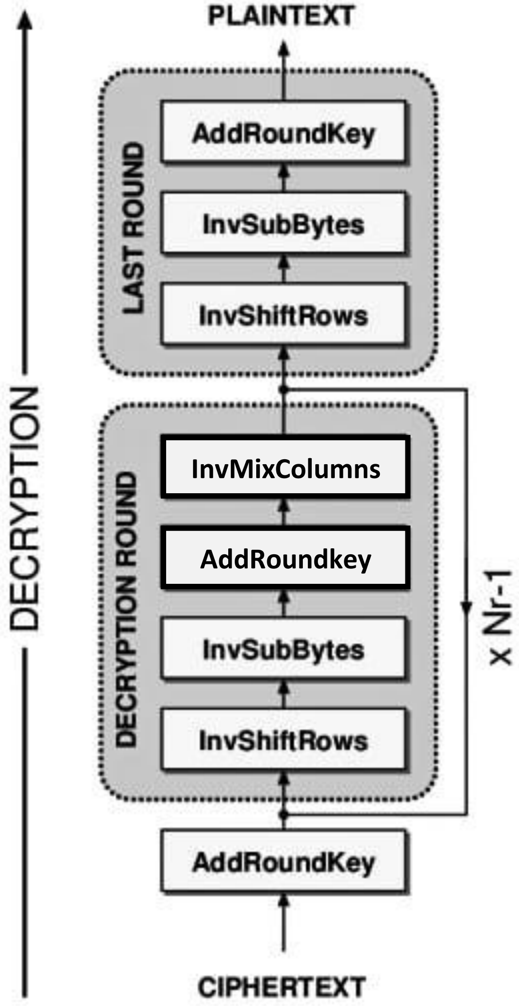


Figure 2.11AES Invers Cipher Function [48]

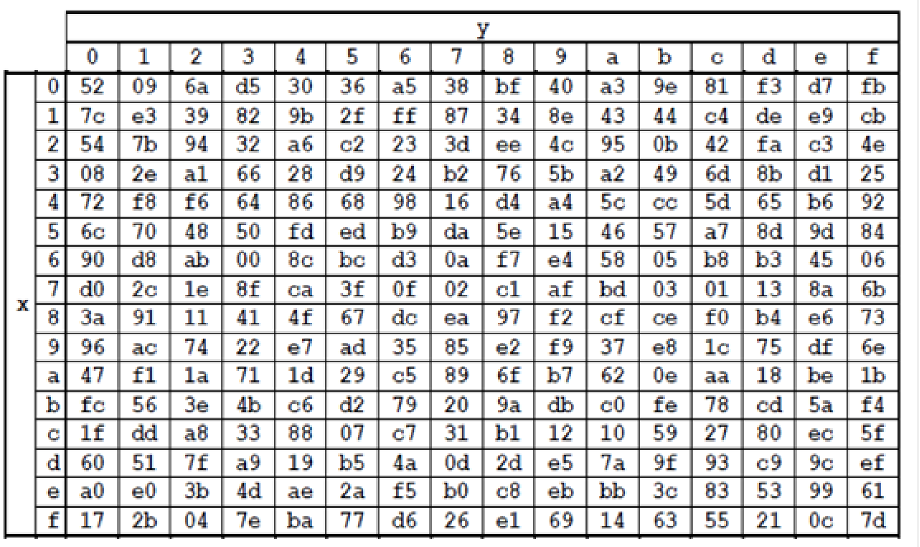
5.1 AES INVERSE CIPHER FUNCTIONS

The AES Inverse Cipher perform has a comparable arrangement of changes as inside the cryptography anyway in the reverse structure, that is, the predefined values that utilized for the each change will be very surprising. During this section, we will discuss concerning every transformations well.

5.1.1InvSubBytes Transformation

InvSubBytes is that the opposite of the PC memory unit replacement change, during which the converse S-Box is applied to each PC memory unit of the State. The inverse S-Box is gift within the Appendix-1 for the reference. The transformation of this method are going to be disbursed within the similar approach as in the SubBytes in the encoding similar to the substitution worth would be determined by the intersection of the row and therefore the column.

For example, if $S_{1,1}$= {53}, then the substitution value would be determined by the intersection of the row with index ‘5’ and the column with index ‘3’. This would result in $S_{1,1}$ having a value of {50}.



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 2.12.INV S-BOX [49]

5.1.2 InvShiftRows Transformation

The InvShiftRows is that the converse of the ShiftRows change. The bytes inside the last 3 columns of the State are consistently moved over totally various quantities of bytes (counterbalances). The primary row, r = 0, is not shifted. The lowest 3 rows are cyclically shifted by Nb - shift(r,Nb) bytes,

Wherever the shift price shift(r,Nb) depends on the row variety. Specifically, the InvShiftRows transformation yield as follows.

$$s'_{r,(c+shift(r,Nb)) \bmod Nb} = s_{r,c} \quad \text{for } 0 < r < 4 \quad \text{and} \quad 0 \le c < Nb$$

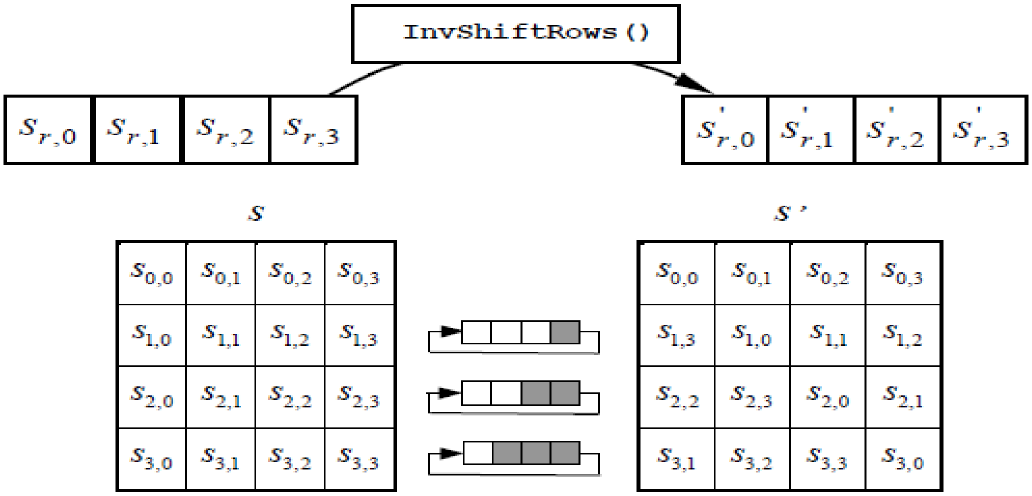

InvShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$
S
S'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,3}$ $s_{1,0}$ $s_{1,1}$ $s_{1,2}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,1}$ $s_{3,2}$ $s_{3,3}$ $s_{3,0}$

Figure 2.13InvShiftRows Operation of the State [50]

The illustration figure will gives the clear view on this InvShiftRows transformation.

5.1.3 InvMixColumns Transformation

The InvMixColumns is that the opposite of the MixColumns change. InvMixColumns works on the State thinking about section by-segment. The pre-characterized 4X4 framework worth and furthermore the first segment of the InvShiftRows state are diagrammatic as follows, for the increase.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

As a result of this multiplication, the four bytes in a column are replaced by the following.

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

Thus, the 4X4 matrix will be obtained which will be given as the input to the next transformation.

5.1.4 Inverse of the AddRoundKey Transformation

The Inverse of the AddRoundKey is similar to the AddRoundKey inside the coding technique. Each part inside the resultant grid of MixColumns and resultant lattice of KeyExpansion are XORed and furthermore the resultant framework of AddRoundKey will be given because the contribution to progressive round.

In this way, all the converse figure changes were referenced on and in the end, the sole factor left to attempt to do is golf shot it at the same time in one inversed primary algorithmic standard. Equally, the forward cipher transformations were combined along to make a round and mixing all the ten Rounds can represent a whole AES coding and cryptography algorithmic rule.

6. Application

AES algorithmic program is one among the premier incredible calculation that are wide used in very surprising fields wherever in the planet. This algorithmic program permits faster than DES and 3DES calculations to code and decode data. moreover, it's utilized in several cryptography protocols love Socket Security Layer (SSL) and Transport Security Layer protocol to produce rather more communications security between consumer and server over the web. Before AES algorithmic program free each of protocols to code and decode information relied on DES algorithm however once showing some vulnerable of this algorithm the web Engineering Task Force (IETF) set to switch DES to AES algorithm. AES can even be found in newest applications and devices that require secret writing practicality: WhatsApp, Facebook traveler and Intel and AMD processor and Cisco devices like router, switch, etc. Additionally, AES sepulcher package is obtainable on several library of package programs love C++ library, C# /.NET, Java and JavaScript that uses to simply and firmly code files from intruders.[33]

7. Advantage and disadvantage of AES

7.1. Advantages of AES

- As it is enforced in each hardware and computer code, it is most sturdy security protocol.
- It uses higher length key sizes love 128, 192 and 256 bits for encoding. Thence it makes AES rule additional sturdy against hacking.
- It is commonest security protocol used for wide varied of applications love wireless communication, money transactions, e-business, encrypted information storage etc.
- It is one in every of the foremost unfold industrial and open supply solutions used everywhere the globe.
- No one will hack your personal data.

- For 128 bit, regarding 2^{128} makes an attempt are required to interrupt. This makes it terribly troublesome to hack it as a result it is very safe protocol.

7.2. Disadvantages of AES:

- It uses too easy pure mathematics structure.
- Every block is often encrypted within the same means.
- Hard to implement with package.
- AES in counter mode is advanced to implement in package taking each performance and security into concerns.

8. Conclusion

Utilizing net and system are expanding quickly. Ordinary heaps of computerized data are trading among clients. Various data is delicate that require to shield from gatecrashers. Encoding calculations assume significant jobs to shield unique data from unapproved access. Varied reasonably algorithms are exist to encipher information. Advanced encoding commonplace (AES) rule is one among the economical algorithm and it is wide supported and adopted on hardware and computer code. This rule permits to cope with totally different key sizes appreciate 128, 192, and 256 bits with 128 bits block cipher. During this paper, explains variety of vital options of AES rule and presents some previous researches that have done thereon to gauge the performance of AES to encipher information below totally different parameters. in step with the results obtained from researches shows that AES has the power to produce way more security compared to alternative algorithms like DES, 3DES etc.

- Basic Terminologies and also the Parameters employed in this rule are mentioned at the sooner section.
- Basic introduction and outline on the AES rule and its prime Level diagram was mentioned.
- Mentioned on AES encoding method, which incorporates AES Cipher Functions and its transformation procedure.
- Steps concerned within the Key enlargement method got.
- AES coding method that incorporates AES Inverse Cipher Functions was explained.

CHAPTER

III

1. Introduction

In this chapter we will expose our work which consists in the development of a system able to perform encryption and decryption using AES algorithm. To demonstrate the proper functioning of the system, we will simulate each bloc among the Round. Then we will follow the steps of encryption process by injection those Inputs:

Clear Text <=x"**32_88_31_e0_43_5a_31_37_f6_30_98_07_a8_8d_a2_34"**;

Key <= x"**2b_28_ab_09_7e_ae_f7_cf_15_d2_15_4f_16_a6_88_3c"**;

And the result of the encrypted sequence is this output:

Cipher text <= x"**39_02_dc_19_25_dc_11_6a_84_09_85_0b_1d_fb_97_32“**;

Besides we will break the output to get the input again so the new inputs are :

Cipher text <= x"**39_02_dc_19_25_dc_11_6a_84_09_85_0b_1d_fb_97_32“**;

Key <= x"**2b_28_ab_09_7e_ae_f7_cf_15_d2_15_4f_16_a6_88_3c"**;

Finally, the new output will be :

Clear Text <=x"**32_88_31_e0_43_5a_31_37_f6_30_98_07_a8_8d_a2_34"**;

Using the FPGA card of Xilinx_VERTEX-7_XC7V585T family, and The VHDL language for modeling of the proposed architectures. The ISE software tools can verify the AES fully and its blocks independently.

2. Introduction to FPGA

During the last decades, FPGAs wore the crown of the digital circuit implementation media kingdom (Field-Programmable Gate Arrays). They have proven their superiority compared to other technologies through their creation lies in their design, that governs the character of their programmable logic practicality and their programmable interconnect. FPGA design features a dramatic result on the standard of the ultimate device’s speed performance, space potency, and power consumption. [29]

2.1. What is FPGA?

The FPGA or field-programmable gate array may be a marvelous technology utilized by electronic system developers to style, debug, and implement distinctive hardware solutions while not having to develop custom silicon devices. Xilinx may be a semiconductor manufacturer of normal FPGA chips that are sold-out blank or unprogrammed to customers. Then, developers involе these devices to implement their own systems. If a feature changes or a bug is discovered, the user will merely load a brand new program to the FPGA to offer a new or upgrade product.

This method will even continue once loading within the variety of microcode upgrades. The act of programming the FPGA is named configuration to discern it from loading any associated software system programs. But with fashionable FPGAs, the road is blurring between hardware configuration and software system programming. [28]

2.2. Purpose of using FPGA

Field Programmable Gate Arrays (FPGAs) are pre-fabricated chemical element devices, which will be electrically programmed to become nearly any reasonably digital circuit or system. Initial FPGA era was virtually introduced two and a half decades past. That FPGA contained terribly tiny range of logic blocks and I/Os. Since then, FPGAs have witnessed a colossal growth each in terms of capability and market. They needed to evolve from an approach where the devices were once thought-about solely as glue logic towards devices that will currently implement complete applications. FPGAs are currently wide used for implementing digital circuits in a very wide range of markets as well as telecommunications, automotive systems and client natural philosophy.

FPGAs carries with it an array of blocks of probably differing kinds, as well as general purpose logic blocks and specific purpose exhausting blocks like memory and number blocks. Among these blocks, general purpose logic blocks are programmable and at the side of specific purpose exhausting blocks they're enclosed by a programmable routing cloth that enables these blocks to be programmably interconnected. The array of blocks at the side of routing cloth is enclosed by programmable input/output blocks that connect the chip to the skin world. The “programmable” term in FPGA indicates that nearly any hardware operate is programmed into it when its fabrication. This customization is realized with the assistance of programming technology, that could be a technique that changes the behavior of the give the sector when its fabrication. [30]

3. Introduction to HDL

The objective of this work is to research the degree to those distinct event hardware description languages, in general, and VHDL (Very High Speed Integrated Circuits Hardware description Language), above all, will model continuous event, or analog, electronic circuits and systems. VHDL, developed at first below the auspices of the us Department of Defense (DoD) and later under the Institute of Electrical and Electronic Engineers (IEEE), is meant to explain the static and dynamic linguistics of digital systems. Digital systems are in the main characterized by the character of the constituent signals; the signals are distinct in time and price. VHDL was developed to explain digital systems, there was never a particular effort to preclude VHDL from additionally describing analog systems. Since VHDL may be a made and powerful language, it's cheap to expect that VHDL is capable of describing attributes of sure sorts of analog systems.

Hence, with the growing importance of mixed analog/digital systems, there's a necessity to look at the pertinency of mistreatment VHDL to model analog behavior. [37]

This study appearance at a broad vary of strategies of characterizing analog systems and presents representative results of mistreatment VHDL to model such systems. The categories of analog systems and therefore the strategies of characterizing their behavior are examined so as of skyrocketing quality, beginning with basic analog ideas and dealing toward network freelance data-sampled analog systems so network-dependent data-sampled systems. As the name implies, network-independent data-sampled analog systems exhibit behaviors, input/output transforms, freelance of their larger atmosphere (networks they will be a region of), whereas network-dependent data-sampled analog systems use info concerning their larger atmosphere in generating input/output transforms. Even easy analog circuits, corresponding to one semiconductor electronic equipment, will exhibit voltage and current gain that are functions of the loading of neighboring circuits either driving or being driven by the electronic equipment. [37]

3.1. A language of HDL?

Hardware description languages (HDL) comparable to VHDL and Verilog have found their ways in virtually each facet of the look of digital hardware systems. Since their origination, they step by step well-tried to be a necessary part of electronic equipment design methodologies and style automation tools, ever surpassing their original goals of being description and simulation languages. Their use for automatic synthesis, formal proof, and testing are sensible examples.

HDLs bring heaps of attention-grabbing options to the look method. Foremost, they are true languages that styleers can perceive and might use to speak design descriptions. They support descriptions of each structural and activity aspects of styles over many abstraction levels from abstract specifications right down to technology dependent netlists.

Additionally, the likelihood to outline many alternatives for a similar style together with the flexibility to develop parameterized models permits to upset the ever-increasing complexness of today's styles. They even have well outlined operational linguistics that build HDL descriptions practicable and verifiable through simulation. Last, but not least, normal HDLs, comparable to VHDL and Verilog, are mostly supported by EDA tool suppliers with the consequence for the users to own an outsized alternative of tools and to make sure ability and style recycle among design tools.

3.2. Why an HDL language?

. Why not use a standard computer language like C/C++, Pascal or ADA?

- Computer languages are not suitable for modeling and simulation of electronic
- Systems because of several things:
- Data types and signals.
- Parallel execution.
- Propagation delays. [31]

Structural programming languages (Pascal, Ada, C) are less usually wont to describe hardware. The result's a "simulating program". At a awfully abstract description stage, designers with applied science backgrounds will use an setting appreciate confab or C++ to explain their systems. The object-oriented programming strategies are inestimably useful in cases wherever the planning drawback is poorly outlined or structured at the start of the project. Declarative languages appreciate logic programming could prove helpful and natural to specific the functionalities of sure quite circuits. Some proprietary languages are specialized to explicit domains appreciate network protocol description, finite-state machines, etc. [32]

3.3.Advantages of an HDL language

- Independent of technologies.
- Reuse of existing designs.
- Hierarchical design.
- Use of libraries .
- Means of exchange between the different design tools.
- Improves the quality.
- Reduces the cost of design.
- Reduces Time **to Market** (TTM) **start-**up time [In trade, the time to market is the time between the design of a product and its sale]. **.** [31]

3.4. Roles of an HDL language.

The technology of translating a given digital style task into digital logic has undergone several changes. The Nineteen Seventies and Eighties witnessed a schematic style approach. From the mid-1990s onward, digital style has been done exploitation hardware description language (HDL). HDLs came into existence to assist the designer with the simulation of digital logic. The provision of synthesis tools that convert high-density lipoprotein logic to FPGA primitives has created the digital style entry methodology of selection. Given the actual fact that HDLs launched primarily as a simulation language, there are several high-density lipoprotein constructs that can't be synthesized to digital logic. Describing a digital design exploitation high-density lipoprotein is typically the primary step toward prototyping the planning using FPGA.

4. VHDL and synthesis

VHDL, which stands for VHSIC Hardware Description Language, describes digital circuits. In simulation, VHDL supply files are analyzed and an outline of the behavior is expressed within the variety of a netlist. A netlist may be a laptop illustration of a set of logic units and the way they're to be connected. The logic units are generally AND/OR/NOT gates or some set of a primitive that produces sense for the target (4-LUTs, for example). The behavior of the circuit is exercised by providing a sequence of inputs. The inputs, known as take a look at vectors, is created manually or by writing a program/script that generates them. The part that's generating take a look at vectors and driving the device beneath test is usually known as a test bench. [33]

Synthesizable VHDL

In VHDL, there are 2 major designs or styles of writing hardware descriptions. each designs are valid VHDL codes; but, they model hardware otherwise. [33]

This impacts synthesis, simulation, and, in some cases, designer productivity. These forms are: Structural/data flow circuits are represented in terms of logic units (either AND/OR/NOT gates or larger practical units akin to multipliers) and signals. knowledge flow could be a kind of structural description that has syntactical support to create it easier to precise formal logic. [33]

Behavioral circuits are represented in an important (procedural) language to explain however the outputs are relating to the inputs as a method. [33]

A third vogue exists as a combination between each structural and behavioral designs. For programmers acquainted with serial processors, the behavioral type of VHDL appears natural. During this vogue, the method being represented is evaluated by “executing the program” within the process block. For this reason, often advanced hardware is expressed compactly and quickly — increasing productivity. It conjointly has the profit that simulations of sure hardware styles are abundant quicker as a result of the method block is dead directly. However, because the style becomes additional advanced, it's attainable to put in writing behavioral descriptions that can't be synthesized. Changing behavioral styles into netlists of logic units is named High-Level Synthesis, bearing on the actual fact that behavioral VHDL is additional abstract (or higher) than structural vogue. [33]

In distinction, as a result of the structural/data-flow vogue describes logic units with celebrated implementations, these VHDL codes nearly always synthesize. Also, collecting massive systems (such as Platform FPGAs) needs structural vogue at the highest level because it is combining large operate units (processors, peripherals, etc.). Likewise, it's price noting that some structural codes don't synthesize well. Associate example of this can be employing a massive RAM in a very hardware style. A RAM isn't troublesome to explain structurally attributable to its easy, repetitive style. However, in simulation, this tends to provide an oversized organization, that has to be traversed on every occasion a sign changes. In distinction, a behavioral model of RAM simulates quickly as a result of it matches the processor's design well. [33]

5. Xilinx ISE Overview

ISE (Integrated Synthesis Environment) is the product programming software Xilinx (CPLD, FPGA Spartan and Virtex…). This tool allows you to create projects with several types of files (HDL, schematic, UCF, EDIF, etc.), to compile, to create implementation constraints with timing constraints on the clocks, to determine pin location and create stimuli files. The ISE Project Navigator provides a design environment and groups all the tools necessary for the design, simulation and implementation of a project such as :

- An editor for texts, diagrams and state diagrams.
- A VHDL and Verilog compiler.
- A simulator.
- Tools for managing time constraints.
- Tools for synthesis.
- Tools for verification.
- Tools for implementation on FPGA and CPLD. [34]

6. Test-bench

One of the characteristics of VHDL is that it permits a verification check bench to be written within the same language because the design to be verified. We have a tendency to embrace the planning to be verified as a part instance in a very check bench model. Moreover, we have a tendency to write VHDL statements to use sequences of check values to the input ports of the planning, and verify that the planning produces the expected output values. However, some aspects of the language that we've seen to date build it exhausting to verify designs. Whereas they're smart for a design in isolation, they'll stop a check bench from accessing things internal to a design. A check bench may have to watch the state of internal signals, or force internal signals to specific values. [35]

Another verification approach depends less on testing and additional on formal proof of correctness of a style. So, as to prove correctness, we want to be ready to specify what the planning is meant to try to. VHDL permits to specific style intent within the kind of properties written in the Property Specification Language (PSL). Whereas PSL may be a separate language, outlined by an IEEE normal and applicable to variety of hardware description languages, VHDL permits to implant PSL specification among a VHDL model. [35]

PSL is that the IEEE normal Property Specification Language (IEEE Std 1850). It permits specification of temporal properties of a model that may be verified either statically (using a proper proof tool) or dynamically (using simulation checkers). VHDL permits PSL code to be embedded as a part of a VHDL model. This makes design for verification a way additional natural activity, and simplifies development and maintenance of models. [35]

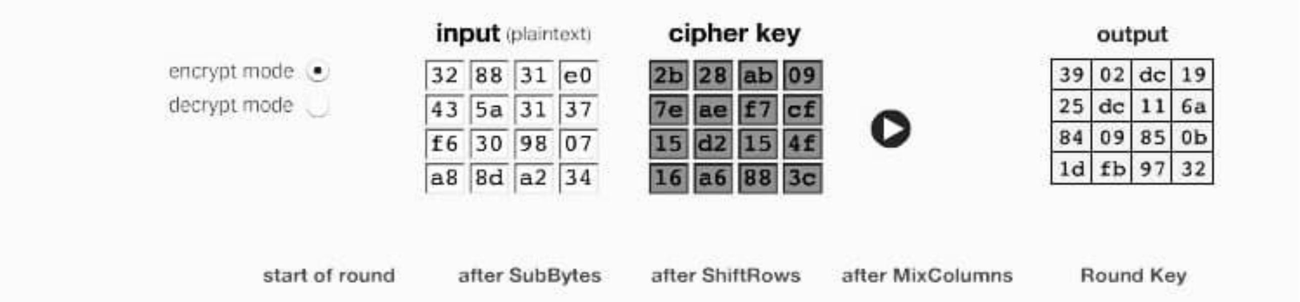
7. Implementing AES in VHDL

The formula of AES encompasses a serious weight within the cryptography word, it play several roles in programing languages like C++ or MATLAB while not forgetting the VHDL.

First of all, we have a tendency to begin by checking the proper functioning of all the blocks of design, through simulations that we have a tendency to administrated victimization the ISE atmosphere.

To do this, we have a tendency to used a similar samples of this following AES-coder-decoder Verify the accuracy of the results delivered by our design. We will throw in the towel the subsequent some samples of simulations of the various blocks constituting our design and the RLT schematics and therefore the chronograms of simulation check bench.

8. The AES CACULATOR ENCYPTION MODE



encrypt mode ●
decrypt mode ○

input (plaintext)

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

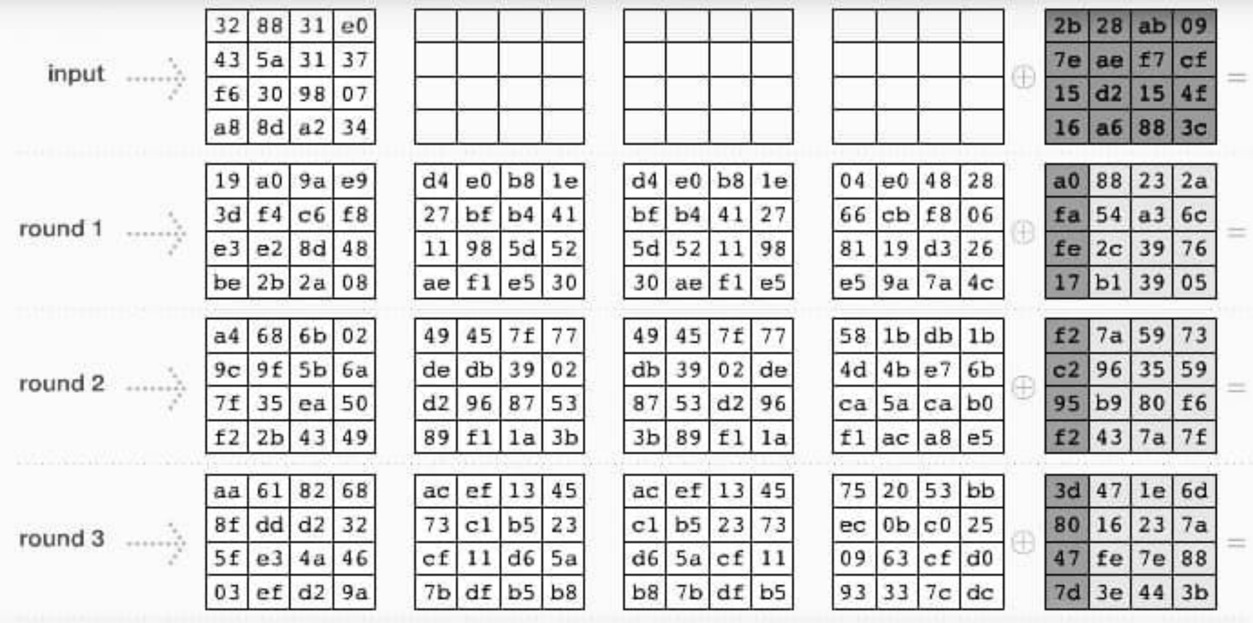
cipher key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

output

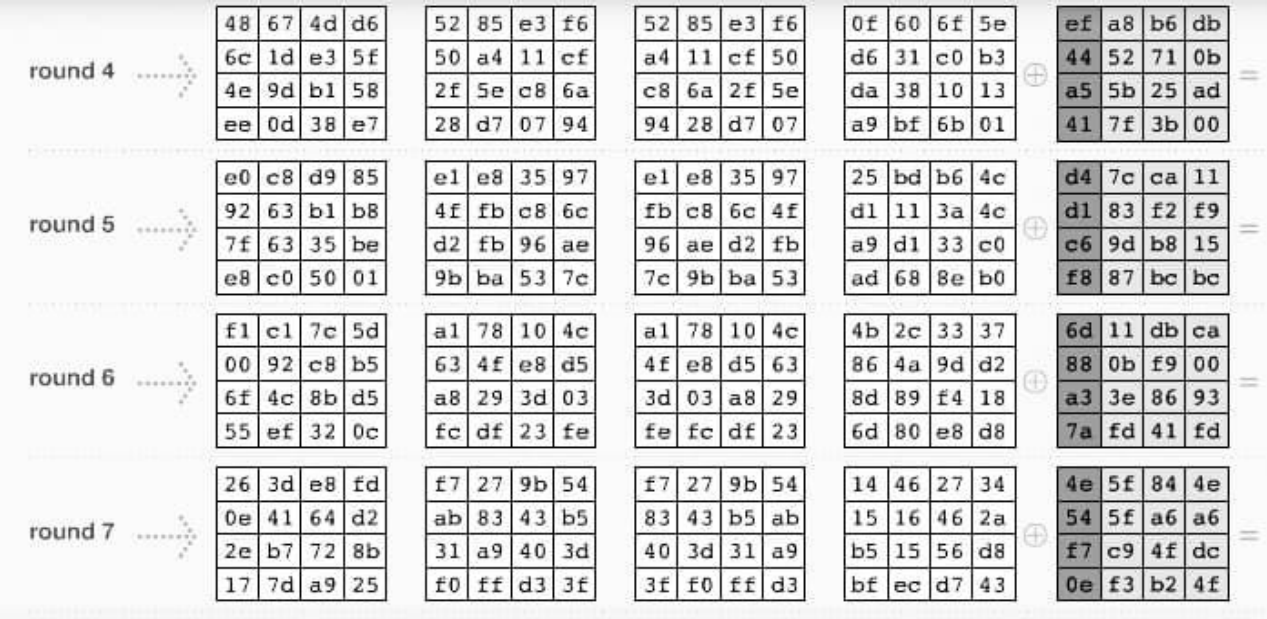
39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

Figure 3.1.1.The input with key and Encryption output [51]



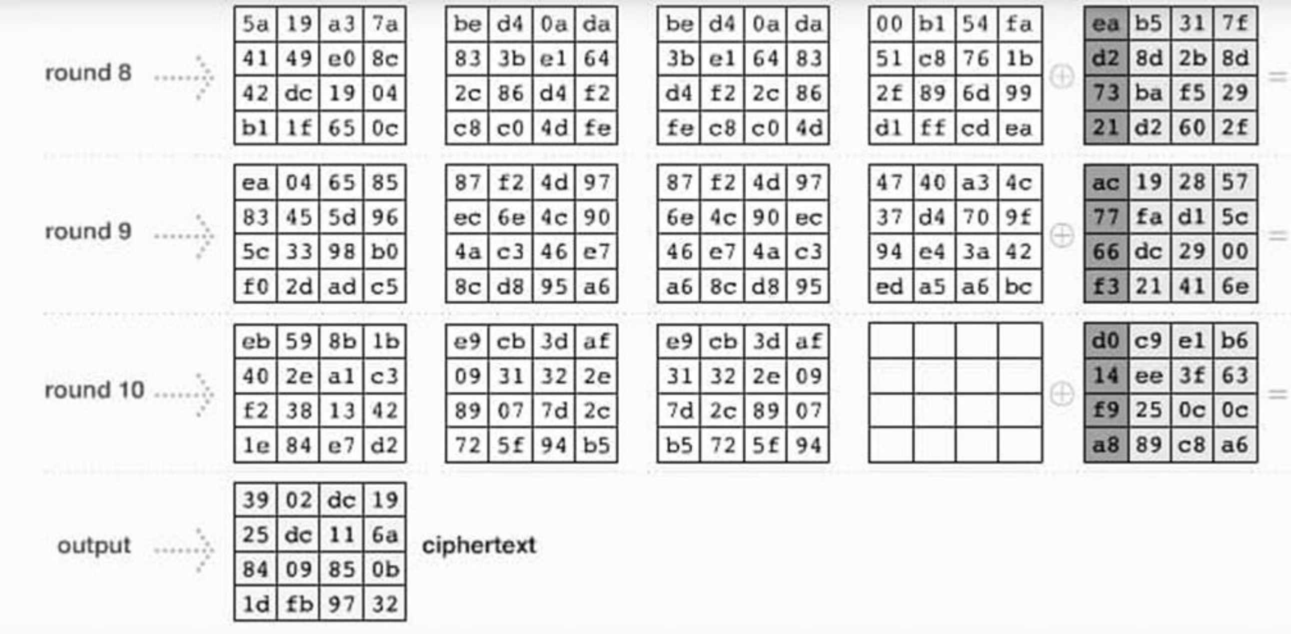
input	32	88	31	e0														2b	28	ab	09	
	43	5a	31	37													⊕	7e	ae	f7	cf	=
	f6	30	98	07														15	d2	15	4f	
	a8	8d	a2	34														16	a6	88	3c	
round 1	19	a0	9a	e9	d4	e0	b8	1e	d4	e0	b8	1e	04	e0	48	28		a0	88	23	2a	
	3d	f4	c6	f8	27	bf	b4	41	bf	b4	41	27	66	cb	f8	06	⊕	fa	54	a3	6c	=
	e3	e2	8d	48	11	98	5d	52	5d	52	11	98	81	19	d3	26		fe	2c	39	76	
	be	2b	2a	08	ae	f1	e5	30	30	ae	f1	e5	e5	9a	7a	4c		17	b1	39	05	
round 2	a4	68	6b	02	49	45	7f	77	49	45	7f	77	58	1b	db	1b		f2	7a	59	73	
	9c	9f	5b	6a	de	db	39	02	db	39	02	de	4d	4b	e7	6b	⊕	c2	96	35	59	=
	7f	35	ea	50	d2	96	87	53	87	53	d2	96	ca	5a	ca	b0		95	b9	80	f6	
	f2	2b	43	49	89	f1	1a	3b	3b	89	f1	1a	f1	ac	a8	e5		f2	43	7a	7f	
round 3	aa	61	82	68	ac	ef	13	45	ac	ef	13	45	75	20	53	bb		3d	47	1e	6d	
	8f	dd	d2	32	73	c1	b5	23	c1	b5	23	73	ec	0b	c0	25	⊕	80	16	23	7a	=
	5f	e3	4a	46	cf	11	d6	5a	d6	5a	cf	11	09	63	cf	d0		47	fe	7e	88	
	03	ef	d2	9a	7b	df	b5	b8	b8	7b	df	b5	93	33	7c	dc		7d	3e	44	3b	

Figure 3.1.2. Encryption rounds (1^{st} to 3^{rd}) with input encryption [51]



round 4	48	67	4d	d6	52	85	e3	f6	52	85	e3	f6	0f	60	6f	5e		ef	a8	b6	db	
	6c	1d	e3	5f	50	a4	11	cf	a4	11	cf	50	d6	31	c0	b3	⊕	44	52	71	0b	=
	4e	9d	b1	58	2f	5e	c8	6a	c8	6a	2f	5e	da	38	10	13		a5	5b	25	ad	
	ee	0d	38	e7	28	d7	07	94	94	28	d7	07	a9	bf	6b	01		41	7f	3b	00	
round 5	e0	c8	d9	85	e1	e8	35	97	e1	e8	35	97	25	bd	b6	4c		d4	7c	ca	11	
	92	63	b1	b8	4f	fb	c8	6c	fb	c8	6c	4f	d1	11	3a	4c	⊕	d1	83	f2	f9	=
	7f	63	35	be	d2	fb	96	ae	96	ae	d2	fb	a9	d1	33	c0		c6	9d	b8	15	
	e8	c0	50	01	9b	ba	53	7c	7c	9b	ba	53	ad	68	8e	b0		f8	87	bc	bc	
round 6	f1	c1	7c	5d	a1	78	10	4c	a1	78	10	4c	4b	2c	33	37		6d	11	db	ca	
	00	92	c8	b5	63	4f	e8	d5	4f	e8	d5	63	86	4a	9d	d2	⊕	88	0b	f9	00	=
	6f	4c	8b	d5	a8	29	3d	03	3d	03	a8	29	8d	89	f4	18		a3	3e	86	93	
	55	ef	32	0c	fc	df	23	fe	fe	fc	df	23	6d	80	e8	d8		7a	fd	41	fd	
round 7	26	3d	e8	fd	f7	27	9b	54	f7	27	9b	54	14	46	27	34		4e	5f	84	4e	
	0e	41	64	d2	ab	83	43	b5	83	43	b5	ab	15	16	46	2a	⊕	54	5f	a6	a6	=
	2e	b7	72	8b	31	a9	40	3d	40	3d	31	a9	b5	15	56	d8		f7	c9	4f	dc	
	17	7d	a9	25	f0	ff	d3	3f	3f	f0	ff	d3	bf	ec	d7	43		0e	f3	b2	4f	

Figure 3.1.3.Encryption rounds (4th to 7th) [51]



round 8	5a	19	a3	7a	be	d4	0a	da	be	d4	0a	da	00	b1	54	fa	⊕	ea	b5	31	7f	=
	41	49	e0	8c	83	3b	e1	64	3b	e1	64	83	51	c8	76	1b		d2	8d	2b	8d	
	42	dc	19	04	2c	86	d4	f2	d4	f2	2c	86	2f	89	6d	99		73	ba	f5	29	
	b1	1f	65	0c	c8	c0	4d	fe	fe	c8	c0	4d	d1	ff	cd	ea		21	d2	60	2f	
round 9	ea	04	65	85	87	f2	4d	97	87	f2	4d	97	47	40	a3	4c	⊕	ac	19	28	57	=
	83	45	5d	96	ec	6e	4c	90	6e	4c	90	ec	37	d4	70	9f		77	fa	d1	5c	
	5c	33	98	b0	4a	c3	46	e7	46	e7	4a	c3	94	e4	3a	42		66	dc	29	00	
	f0	2d	ad	c5	8c	d8	95	a6	a6	8c	d8	95	ed	a5	a6	bc		f3	21	41	6e	
round 10	eb	59	8b	1b	e9	cb	3d	af	e9	cb	3d	af					⊕	d0	c9	e1	b6	=
	40	2e	a1	c3	09	31	32	2e	31	32	2e	09						14	ee	3f	63	
	f2	38	13	42	89	07	7d	2c	7d	2c	89	07						f9	25	0c	0c	
	1e	84	e7	d2	72	5f	94	b5	b5	72	5f	94						a8	89	c8	a6	
output	39	02	dc	19	ciphertext																	
	25	dc	11	6a																		
	84	09	85	0b																		
	1d	fb	97	32																		

Figure3.1.4. Encryption rounds (8th to 10th) with output [51]

8.1. Key Schedules

The process of generating keys is a module, which has one input Key. It should be noted that key0 to key10 are out-input because each out it used as input to generate the next key.

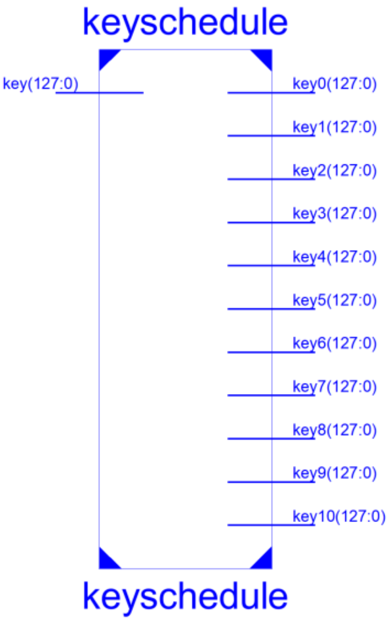


Figure3.2.1.RLT Schematic Key Schedules

The key schedules RLT Schematic show that there is one input which is the key of 16 bytes (128bits); that input will be affected directly to the in_output and will be used to AddRoundKey before starting the ten rounds .The concept in **Figure 2.11**

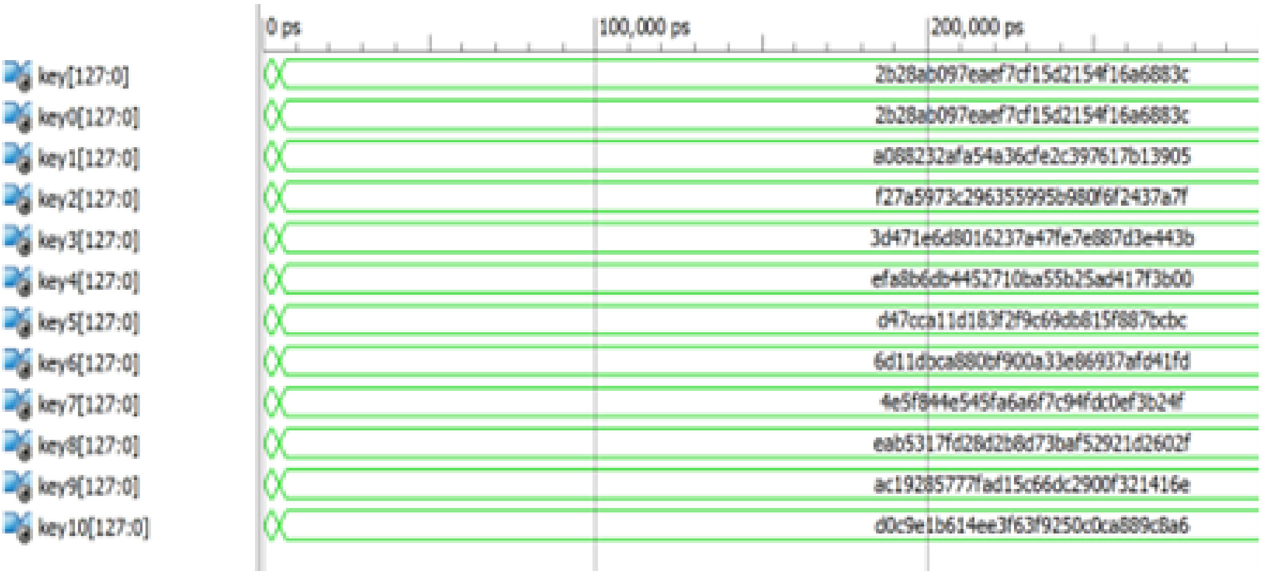
Encryption Keys distribution

- The key0 has used to obtain key1 that used in round 1.
- The key1 has used to obtain key2 that used in round 2.
- The key2 has used to obtain key3 that used in round 3.
- The key3 has used to obtain key4 that used in round 4.
- The key4 has used to obtain key5 that used in round 5.
- The key5 has used to obtain key6 that used in round 6.
- The key6 has used to obtain key7 that used in round 7.
- The key7 has used to obtain key8 that used in round 8.
- The key8 has used to obtain key9 that used in round 9.
- The key9 has used to obtain key10 in the last Round (10th round).

Decryption Keys distribution

- The key0 has used in the last Round 10.
- The key1 has used in round 9.
- The key2 has used in round 8.
- The key3 has used in round 7.
- The key4 has used in round 6.
- The key5 has used in round 5.
- The key6 has used in round 4.
- The key7 has used in round 3.
- The key8 has used in round 2.
- The key9 has used in round 1.
- The key10 has used in round 0.

NOTE: All the keys (1st to 10th) are 16 bytes (128 bits).



key[127:0] 2b28ab097eaef7cf15d2154f16a6883c
key0[127:0] 2b28ab097eaef7cf15d2154f16a6883c
key1[127:0] a088232afa54a36cfe2c397617b13905
key2[127:0] f27a5973c296355995b980f6f2437a7f
key3[127:0] 3d471e6d8016237a47fe7e887d3e443b
key4[127:0] efa8b6db4452710ba55b25ad417f3b00
key5[127:0] d47cca11d183f2f9c69db815f887bcbc
key6[127:0] 6d11dbca880bf900a33e86937afd41fd
key7[127:0] 4e5f844e545fa6a6f7c94fdc0ef3b24f
key8[127:0] eab5317fd28d2b8d73baf52921d2602f
key9[127:0] ac19285777fad15c66dc2900f321416e
key10[127:0] d0c9e1b614ee3f63f9250c0ca889c8a6

Figure3.2.2.Simulation of Key Schedules

8.2. Add Round Key

This operation is carried out by a simple XOR between tow inputs of 128 bit

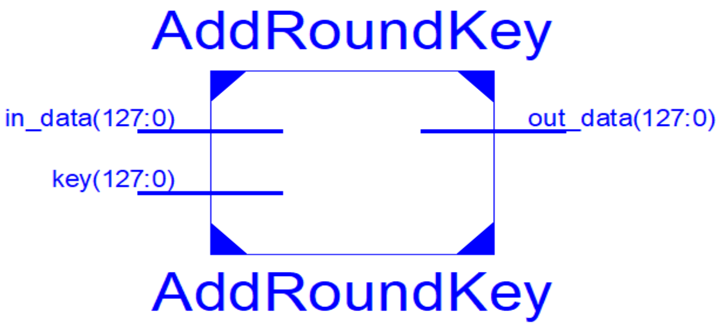
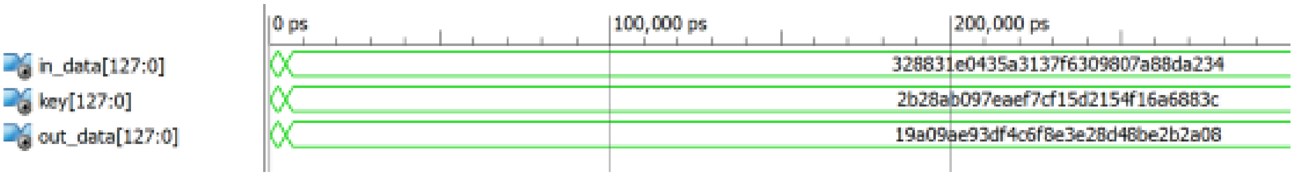


Figure3.3.1.RLT Schematic Add Round Key

The AddRoundKey RLT Schematic show that there is only two inputs (in_data , key) ,whose Xored one by one bytes respectively as the concept explained in Figure 2.4 and Figure 2.5. The result will appear in the output (out_data).



in_data[127:0] 328831e0435a3137f6309807a88da234
key[127:0] 2b28ab097eaef7cf15d2154f16a6883c
out_data[127:0] 19a09ae93df4c6f8e3e28d48be2b2a08

Figure3.3.2.Simulation of Add Round Key

8.1.2 .Sub Bytes

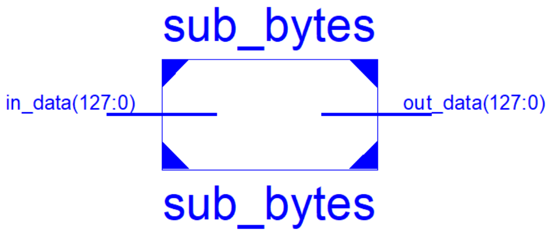


Figure3.4.1.RLT Schematic Sub Bytes

The RLT Schematic above (its concept was well detailed in the previous chapter Figure 2.7 and Figure 2.8) expresses the sub_bytes bloc, which has only one input (in_data) with size of 128 bits. That input (in_data) will be subtracted with S_Box block byte by byte. Figure 3.4.2 shows the result of the output (out_data).

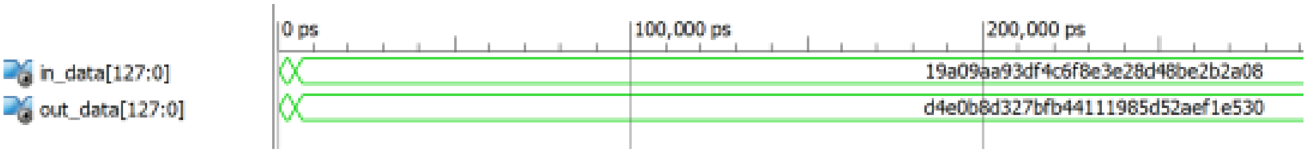

0 ps
100,000 ps
200,000 ps
in_data[127:0]
19a09aa93df4c6f8e3e28d48be2b2a08
out_data[127:0]
d4e0b8d327bfb44111985d52aef1e530

Figure3.4.2.Simulation of Sub Bytes

8.1.3. Shift Rows

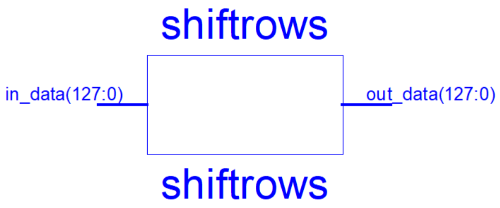


Figure3.5.1.RLT Schematic Shift Rows

The steps in Figure2.9 describe the functionality of this process that showing as the normal state in the input which is shifted and going out the block in the out_data (output).

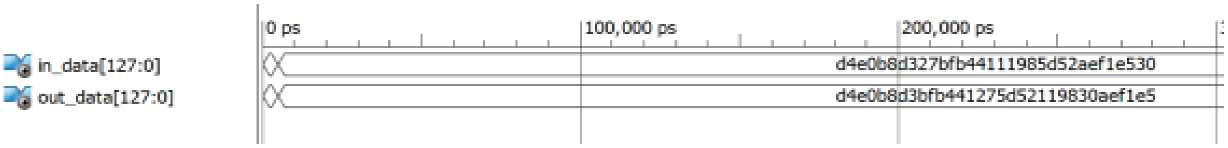

0 ps
100,000 ps
200,000 ps
in_data[127:0]
d4e0b8d327bfb44111985d52aef1e530
out_data[127:0]
d4e0b8d3bfb441275d52119830aef1e5

Figure3.5.2.Simulation of Shift Rows

8.1.4. Mix Columns

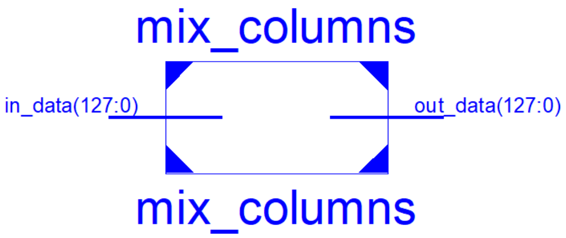


Figure3.6.1.RLT Schematic Mix Columns

As explained in Figure 2.10, the mix_columns block will mix each bytes among the 16 bytes of the in_data(input) , the end of this process showing as a mixed 16 bytes in out_data (output).

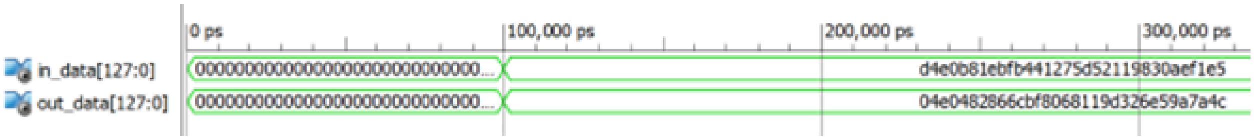


Figure3.6.2.Simulation of Mix Columns

8.1.5. Round (1st to 9th)

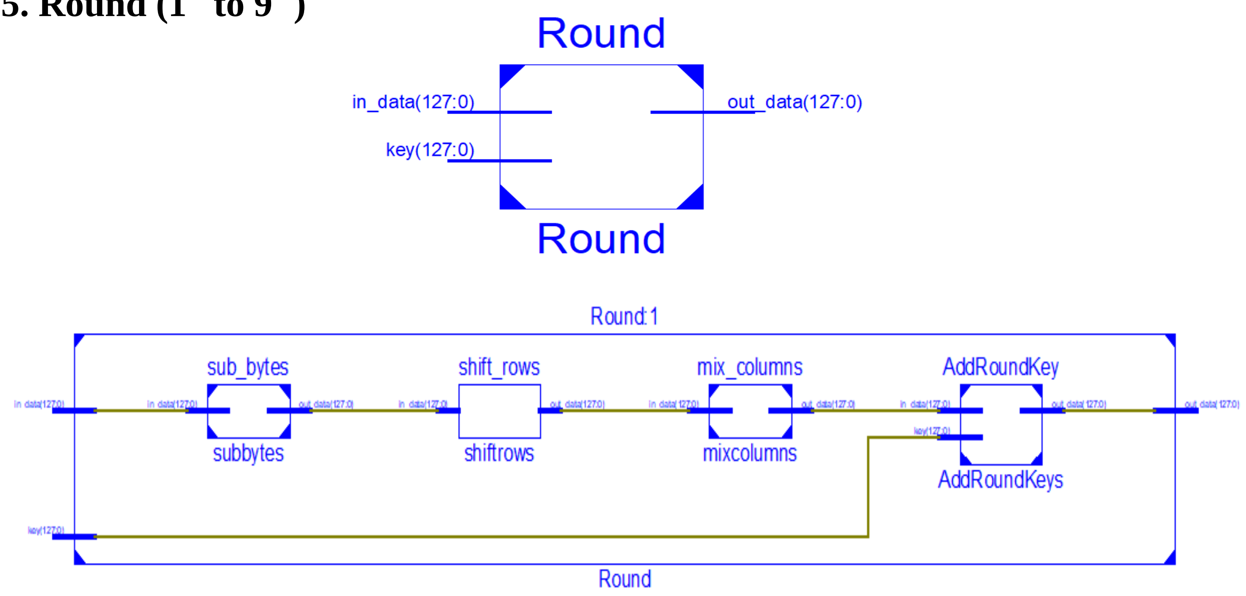


Figure3.7.1.RLT Schematic Round (1st to 9th)

This Block is the most important step to make the AES encryption. It makes the combination of the previous blocks.

The input Round block, is the same input of subbyets block besides the output of the last one is the same input of shiftrows block that will connect its output with the input of mixcolumns block. The output of the last block will be the input of the last step which is addroundkey with the input (mix- columns(out_data)). The other input (key) is coming from the same input of Round (key).

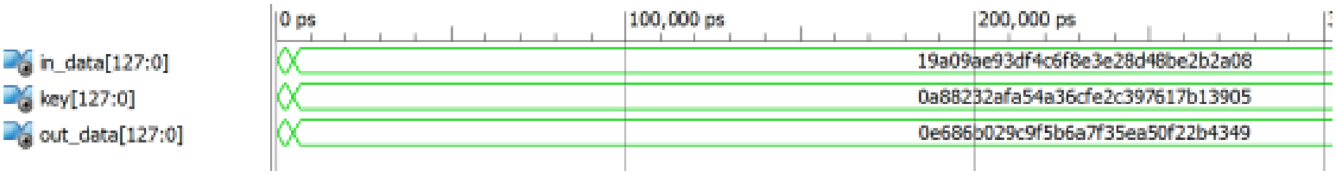

0 ps
100,000 ps
200,000 ps
in_data[127:0]
key[127:0]
out_data[127:0]
19a09ae93df4c6f8e3e28d48be2b2a08
0a88232afa54a36cfe2c397617b13905
0e686b029c9f5b6a7f35ea50f22b4349

Figure3.7.2.Simulation of Round (1st to 9th)

8.1.6 Last Round (10th)

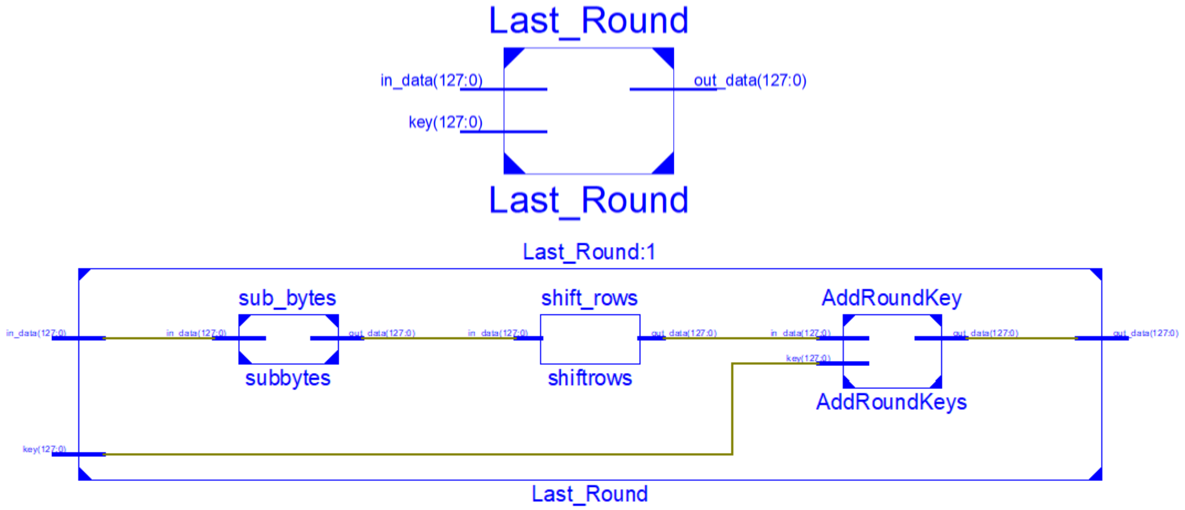
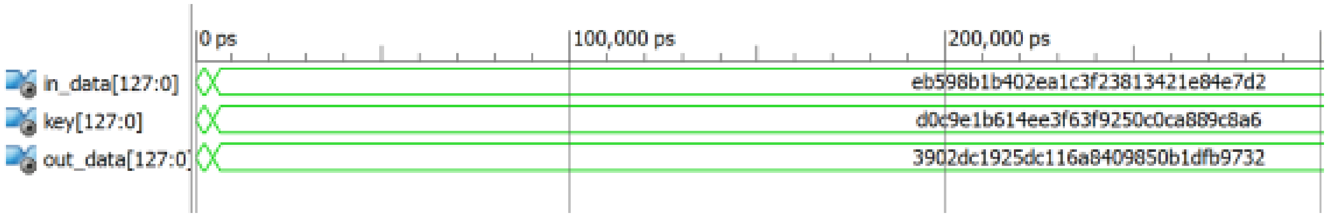


Figure3.8.1.RLT Schematic Last Round (10th)

The last round with the in_data and key (inputs) it's the same principal of the Round(1st to9th) process only without the mixcolumns block.



eb598b1b402ea1c3f23813421e84e7d2

d0c9e1b614ee3f63f9250c0ca889c8a6

3902dc1925dc116a8409850b1dfb9732

Figure3.8.2.Simulation of Last Round (10th)

8.1.7. Encryption

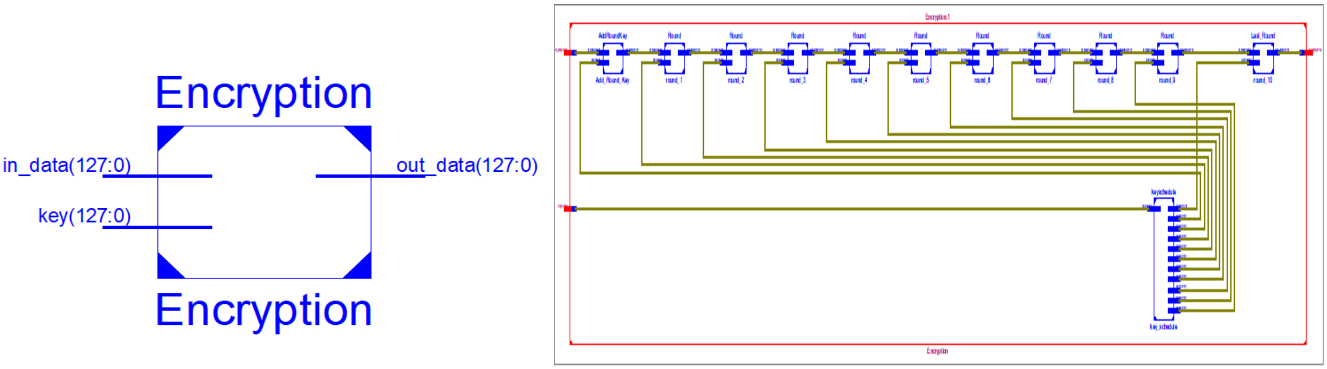
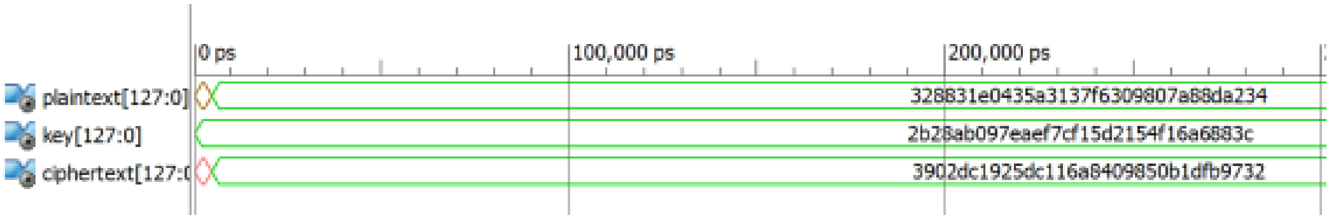


Figure3.9.1.RLT Schematic encryption

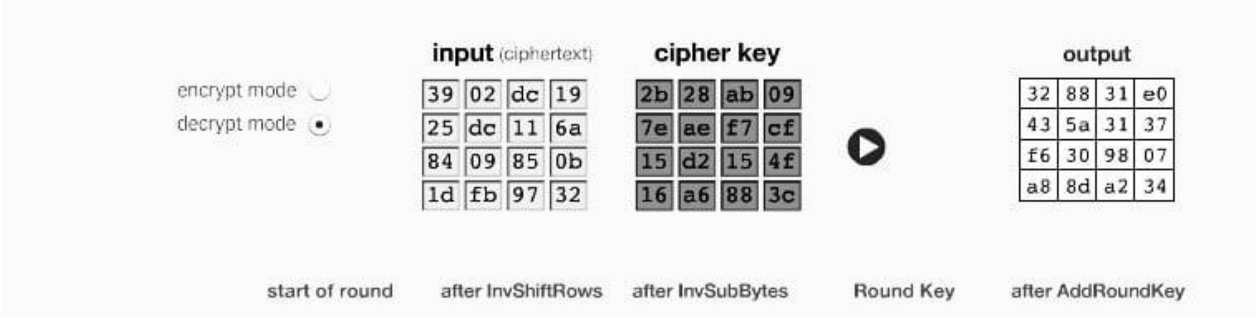
The RLT Schematic shows the 9 rounds and the last one with the key generator, which generates 10 keys each one will be used during its round. The distribution of the keys was well explained in Key Schedules process. The input of encryption process is the same of the first round, the key input of coding process will be the same of key schedule process. Key schedule will manage the keys from the first to the tenth for the rounds with the same ordering. Each output of any round will be the input of the following one respecting Keyes's management .the last round's output is the same of the encryptor.



plaintext[127:0]	328831e0435a3137f6309807a88da234
key[127:0]	2b28ab097eaef7cf15d2154f16a6883c
ciphertext[127:0	3902dc1925dc116a8409850b1dfb9732

Figure3.9.2.Simulation of encryption

9. The AES CACULATOR DECRYPTION MODE



input (ciphertext)

39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

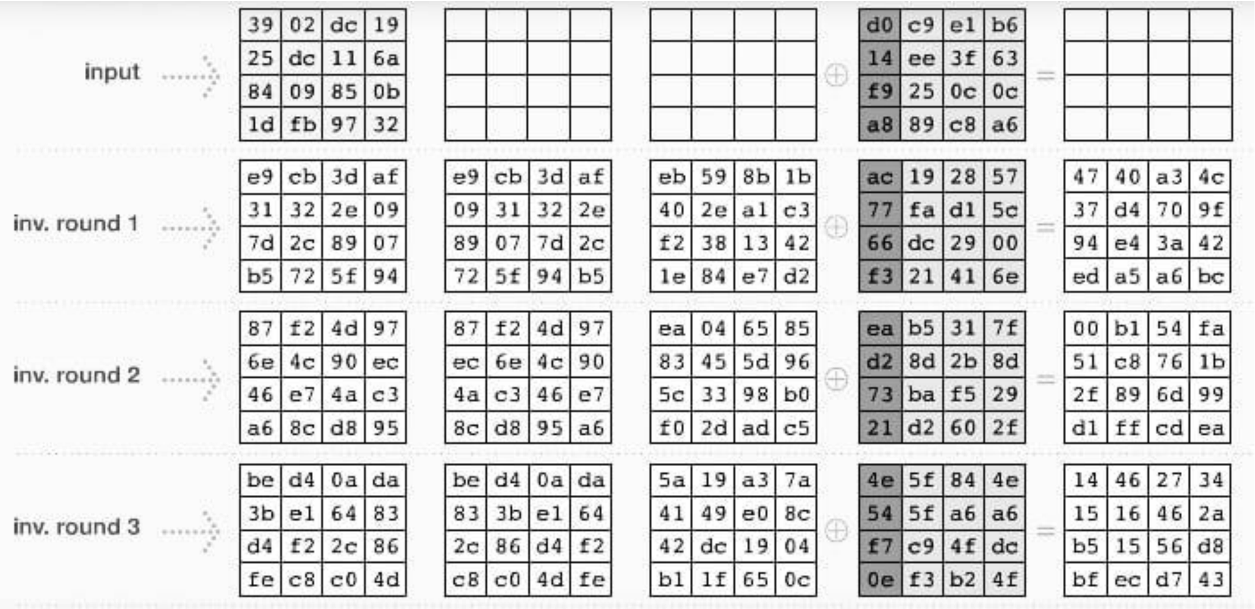
cipher key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

output

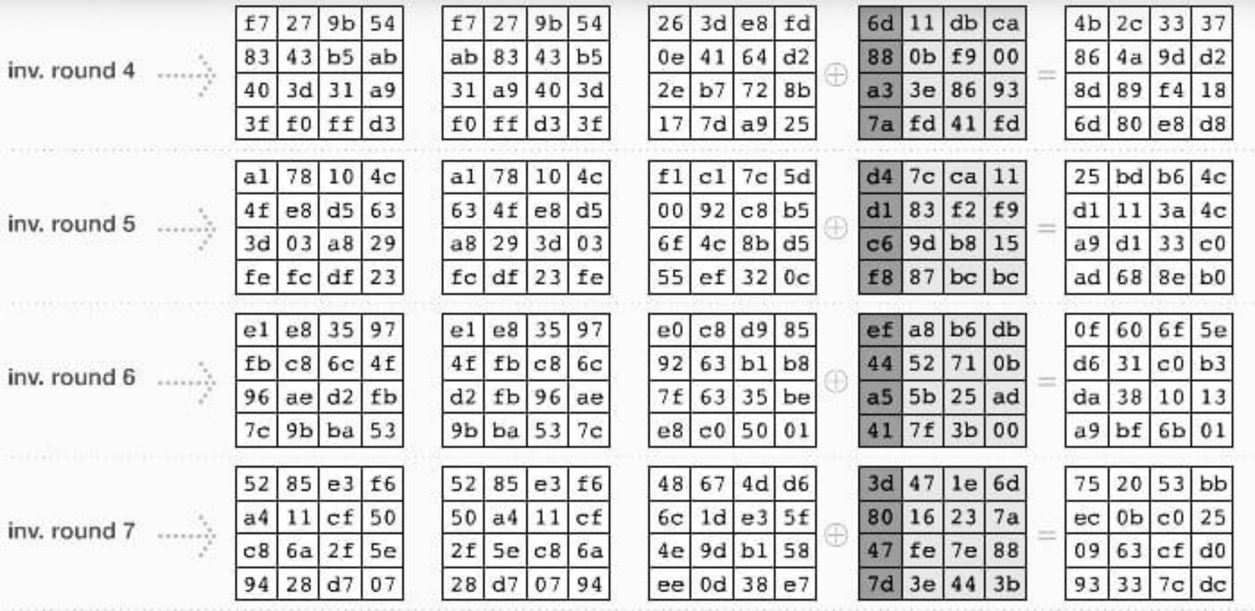
32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Figure3.10.1.The input with key and Decryption output [51]



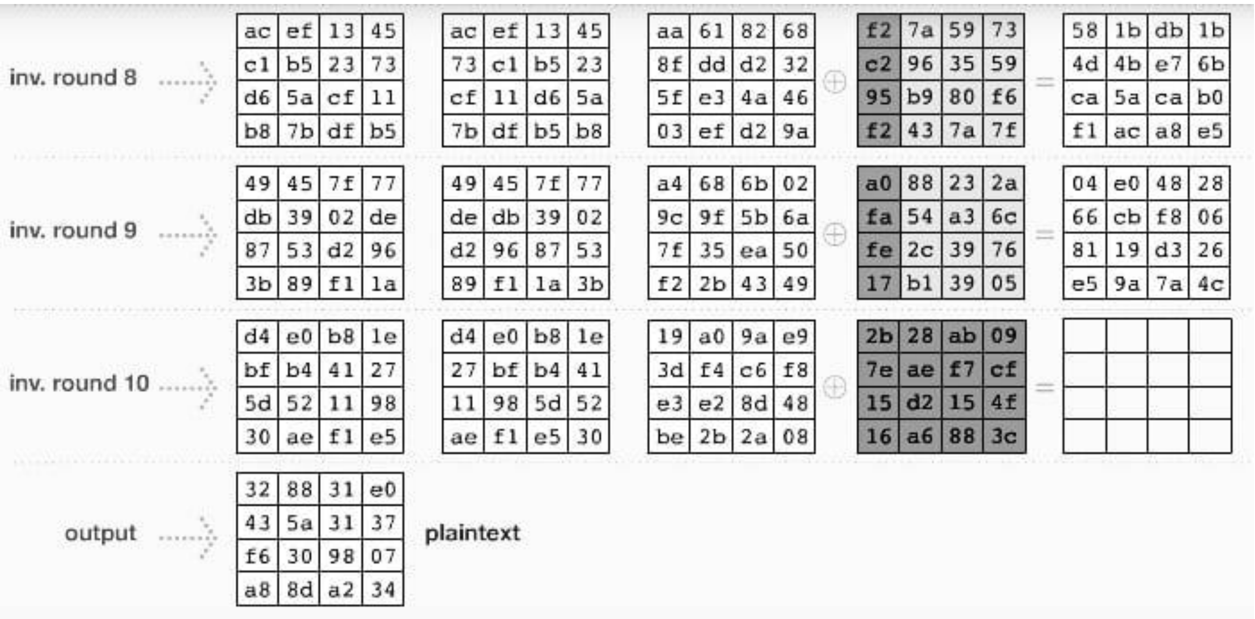
input	39	02	dc	19									⊕	d0	c9	e1	b6	=				
	25	dc	11	6a										14	ee	3f	63					
	84	09	85	0b										f9	25	0c	0c					
	1d	fb	97	32										a8	89	c8	a6					
inv. round 1	e9	cb	3d	af	e9	cb	3d	af	eb	59	8b	1b	⊕	ac	19	28	57	=	47	40	a3	4c
	31	32	2e	09	09	31	32	2e	40	2e	a1	c3		77	fa	d1	5c		37	d4	70	9f
	7d	2c	89	07	89	07	7d	2c	f2	38	13	42		66	dc	29	00		94	e4	3a	42
	b5	72	5f	94	72	5f	94	b5	1e	84	e7	d2		f3	21	41	6e		ed	a5	a6	bc
inv. round 2	87	f2	4d	97	87	f2	4d	97	ea	04	65	85	⊕	ea	b5	31	7f	=	00	b1	54	fa
	6e	4c	90	ec	ec	6e	4c	90	83	45	5d	96		d2	8d	2b	8d		51	c8	76	1b
	46	e7	4a	c3	4a	c3	46	e7	5c	33	98	b0		73	ba	f5	29		2f	89	6d	99
	a6	8c	d8	95	8c	d8	95	a6	f0	2d	ad	c5		21	d2	60	2f		d1	ff	cd	ea
inv. round 3	be	d4	0a	da	be	d4	0a	da	5a	19	a3	7a	⊕	4e	5f	84	4e	=	14	46	27	34
	3b	e1	64	83	83	3b	e1	64	41	49	e0	8c		54	5f	a6	a6		15	16	46	2a
	d4	f2	2c	86	2c	86	d4	f2	42	dc	19	04		f7	c9	4f	dc		b5	15	56	d8
	fe	c8	c0	4d	c8	c0	4d	fe	b1	1f	65	0c		0e	f3	b2	4f		bf	ec	d7	43

Figure3.10.2. Decryption rounds (1st to 3rd) with input encryption [51]



inv. round 4	f7	27	9b	54	f7	27	9b	54	26	3d	e8	fd	⊕	6d	11	db	ca	=	4b	2c	33	37
	83	43	b5	ab	ab	83	43	b5	0e	41	64	d2		88	0b	f9	00		86	4a	9d	d2
	40	3d	31	a9	31	a9	40	3d	2e	b7	72	8b		a3	3e	86	93		8d	89	f4	18
	3f	f0	ff	d3	f0	ff	d3	3f	17	7d	a9	25		7a	fd	41	fd		6d	80	e8	d8
inv. round 5	a1	78	10	4c	a1	78	10	4c	f1	c1	7c	5d	⊕	d4	7c	ca	11	=	25	bd	b6	4c
	4f	e8	d5	63	63	4f	e8	d5	00	92	c8	b5		d1	83	f2	f9		d1	11	3a	4c
	3d	03	a8	29	a8	29	3d	03	6f	4c	8b	d5		c6	9d	b8	15		a9	d1	33	c0
	fe	fc	df	23	fc	df	23	fe	55	ef	32	0c		f8	87	bc	bc		ad	68	8e	b0
inv. round 6	e1	e8	35	97	e1	e8	35	97	e0	c8	d9	85	⊕	ef	a8	b6	db	=	0f	60	6f	5e
	fb	c8	6c	4f	4f	fb	c8	6c	92	63	b1	b8		44	52	71	0b		d6	31	c0	b3
	96	ae	d2	fb	d2	fb	96	ae	7f	63	35	be		a5	5b	25	ad		da	38	10	13
	7c	9b	ba	53	9b	ba	53	7c	e8	c0	50	01		41	7f	3b	00		a9	bf	6b	01
inv. round 7	52	85	e3	f6	52	85	e3	f6	48	67	4d	d6	⊕	3d	47	1e	6d	=	75	20	53	bb
	a4	11	cf	50	50	a4	11	cf	6c	1d	e3	5f		80	16	23	7a		ec	0b	c0	25
	c8	6a	2f	5e	2f	5e	c8	6a	4e	9d	b1	58		47	fe	7e	88		09	63	cf	d0
	94	28	d7	07	28	d7	07	94	ee	0d	38	e7		7d	3e	44	3b		93	33	7c	dc

Figure3.10.3. Decryption rounds (4th to 7th) [51]



inv. round 8	ac	ef	13	45	ac	ef	13	45	aa	61	82	68	⊕	f2	7a	59	73	=	58	1b	db	1b
	c1	b5	23	73	73	c1	b5	23	8f	dd	d2	32		c2	96	35	59		4d	4b	e7	6b
	d6	5a	cf	11	cf	11	d6	5a	5f	e3	4a	46		95	b9	80	f6		ca	5a	ca	b0
	b8	7b	df	b5	7b	df	b5	b8	03	ef	d2	9a		f2	43	7a	7f		f1	ac	a8	e5
inv. round 9	49	45	7f	77	49	45	7f	77	a4	68	6b	02	⊕	a0	88	23	2a	=	04	e0	48	28
	db	39	02	de	de	db	39	02	9c	9f	5b	6a		fa	54	a3	6c		66	cb	f8	06
	87	53	d2	96	d2	96	87	53	7f	35	ea	50		fe	2c	39	76		81	19	d3	26
	3b	89	f1	1a	89	f1	1a	3b	f2	2b	43	49		17	b1	39	05		e5	9a	7a	4c
inv. round 10	d4	e0	b8	1e	d4	e0	b8	1e	19	a0	9a	e9	⊕	2b	28	ab	09	=				
	bf	b4	41	27	27	bf	b4	41	3d	f4	c6	f8		7e	ae	f7	cf					
	5d	52	11	98	11	98	5d	52	e3	e2	8d	48		15	d2	15	4f					
	30	ae	f1	e5	ae	f1	e5	30	be	2b	2a	08		16	a6	88	3c					
output	32	88	31	e0	plaintext																	
	43	5a	31	37																		
	f6	30	98	07																		
	a8	8d	a2	34																		

Figure3.10.4. Decryption rounds (8th to 10th) with output [51]

9.1.1. Inv Shift Rows

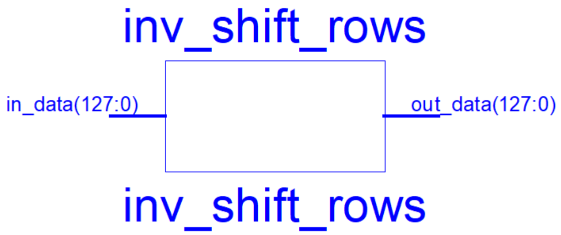


Figure3.11.1.RLT Schematic inv Shift Rows

The steps in Figure 2.13 describe the functionality of this process. These steps showed as the normal state in the input which is shifted and going out the block in the out_data (output).

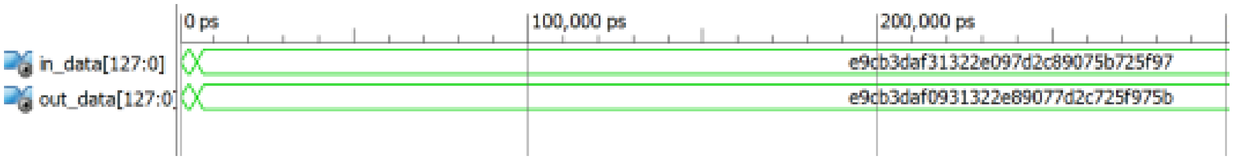

0 ps
100,000 ps
200,000 ps
in_data[127:0]
e9cb3daf31322e097d2c89075b725f97
out_data[127:0]
e9cb3daf0931322e89077d2c725f975b

Figure3.11.2.Simulation of inv Shift Rows

9.1.2 .Inv Sub Byte

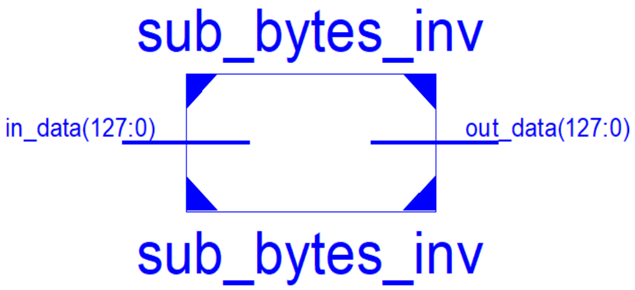


Figure3.12.1.RLT Schematic inv Sub Bytes

The concept exposed in Fig-ure 2.7 and Figure 2.13 is modeled by the RLT Schematic above. With only one input (in_data) with size of 128 bits, which will be subtracted byte by byte. The result is displayed in the output (out_data).

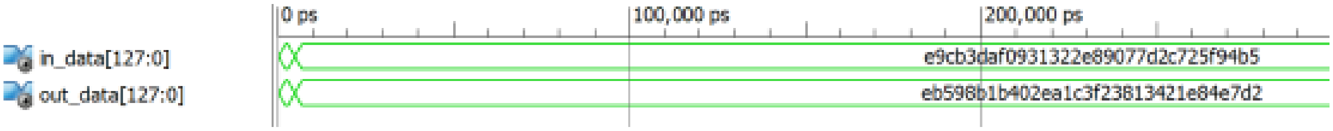
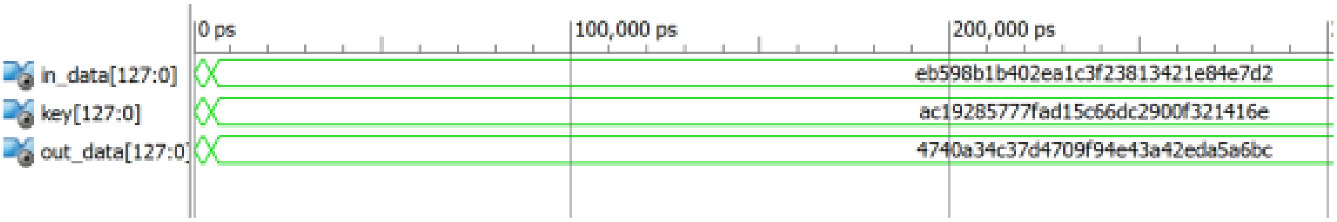

0 ps
100,000 ps
200,000 ps
in_data[127:0]
e9cb3daf0931322e89077d2c725f94b5
out_data[127:0]
eb598b1b402ea1c3f23813421e84e7d2

Figure3.12.2.Simulation of inv Sub Bytes

9.1.3. Inv Mix Columns and inv Add Round Key

9.1.3.1. Inv Add Round Key

Inv addroundkey is the do similar functionality as addroundkey. With only two inputs (in_data , key), that are going to be Xored one by one bytes respectively as explained in the concept in Figure 2.4 and Figure 2.5. The result will be appearing in the output (out_data).



```
              0 ps                 100,000 ps                200,000 ps
in_data[127:0]                        eb598b1b402ea1c3f23813421e84e7d2
key[127:0]                            ac19285777fad15c66dc2900f321416e
out_data[127:0]                       4740a34c37d4709f94e43a42eda5a6bc
```

Figure3.13.1.Simulation of add round key

9.1.3.2. Inv Mix Columns

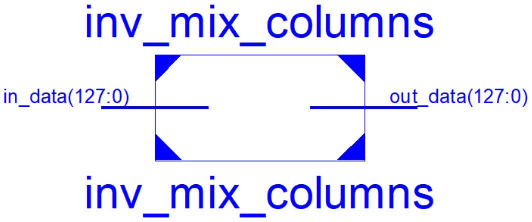
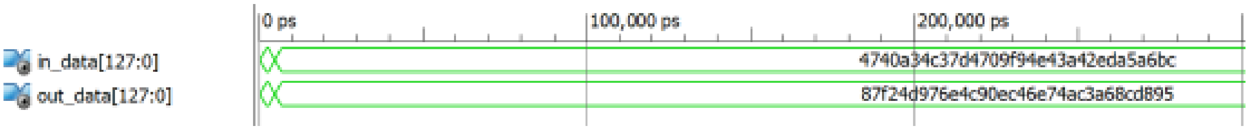


Figure3.13.2.RLT Schematic inv Mix Columns

The concept of this block which was explained in Figure 2.10 and (5.1.3 InvMixColumns Transformation) will reverse the mixing of each bytes among the 16 bytes of the in_data (input). The end of this process showing as an immixed 16 bytes in out_data (output).



```
              0 ps                 100,000 ps                200,000 ps
in_data[127:0]                        4740a34c37d4709f94e43a42eda5a6bc
out_data[127:0]                       87f24d976e4c90ec46e74ac3a68cd895
```

Figure3.13.3.Simulation of inv Mix Columns

9.1.4. Inv Round (1^{st} to 9th)

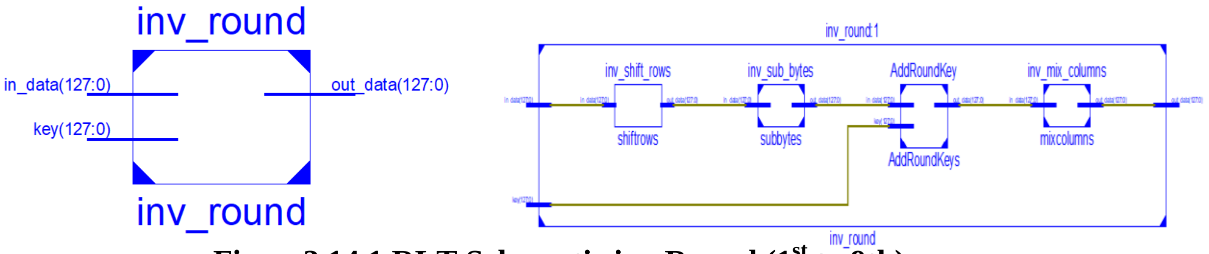
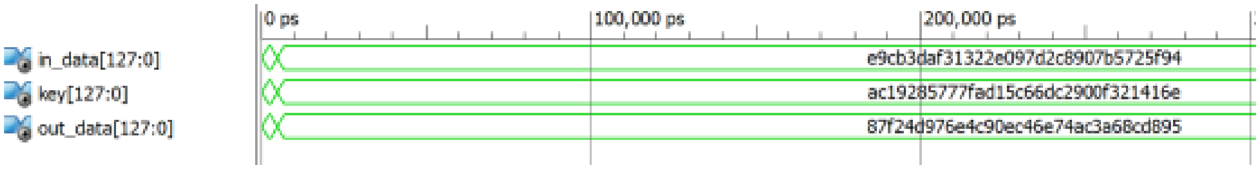


Figure3.14.1.RLT Schematic inv Round (1^{st} to 9th)

This Block is the most important step to make the AES body in decryption. It is the combination of the previous blocks receptively the input Round block, is the same input of inv-shiftrows block besides the output of the last one is the same input of inv-subbytes block that will connect its output with the input of addroundkey block plus the other input (key) is coming from the same input of inv-Round (key). Once more, the output of the last one it will be the input of the last step which is inv-mixcolumns with the input (addroundkey (out_data)) .



```
            |0 ps        |100,000 ps        |200,000 ps
in_data[127:0]     e9cb3daf31322e097d2c8907b5725f94
key[127:0]         ac19285777fad15c66dc2900f321416e
out_data[127:0]    87f24d976e4c90ec46e74ac3a68cd895
```

Figure3.14.2.Simulation of inv Round(1st to 9th)

9.1.5. Inv Last Round (10th)

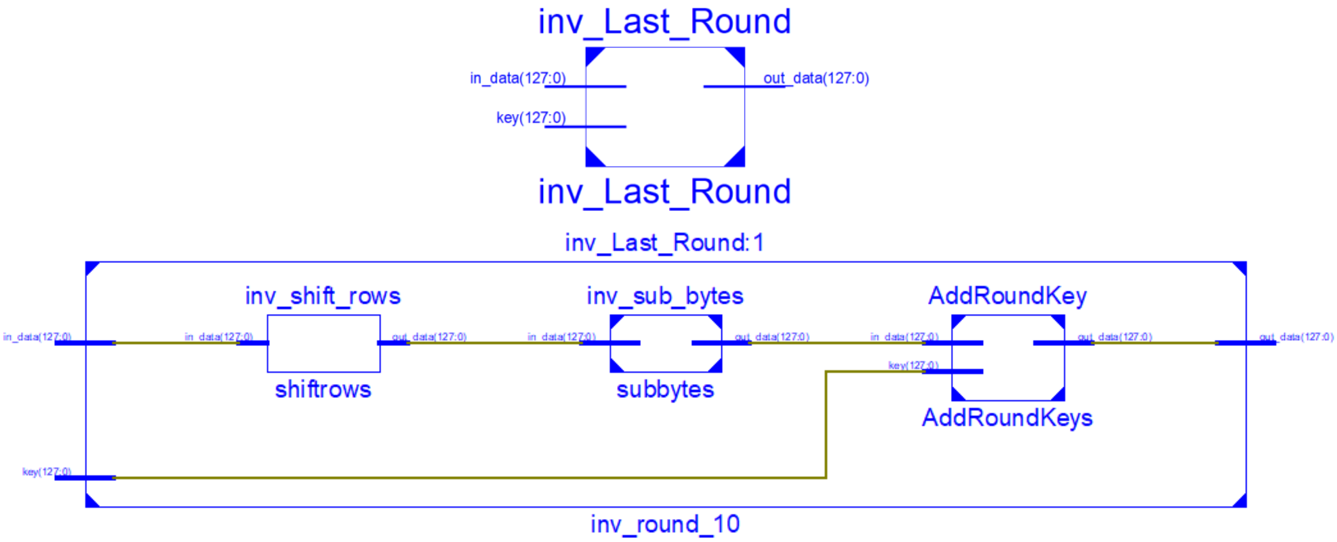
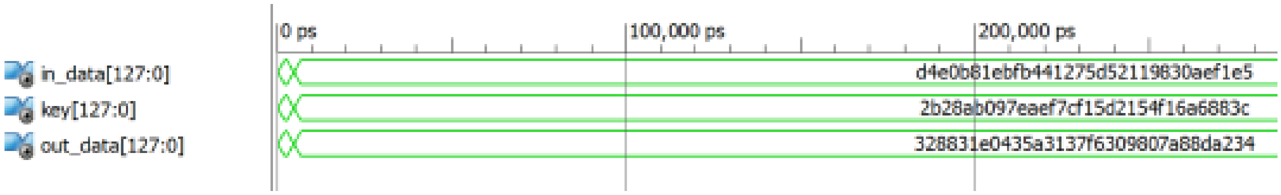


Figure3.15.1.RLT Schematic inv Last Round(10th)

The last round with the in_data and key (inputs) it's the same principal of the inv-Round (1st to 9th) process without the inv- mixcolumns block.



```
            |0 ps        |100,000 ps        |200,000 ps
in_data[127:0]     d4e0b81ebfb441275d52119830aef1e5
key[127:0]         2b28ab097eaef7cf15d2154f16a6883c
out_data[127:0]    328831e0435a3137f6309807a88da234
```

Figure3.15.2.Simulation of inv Last Round (10th)

9.1.6. Decryption

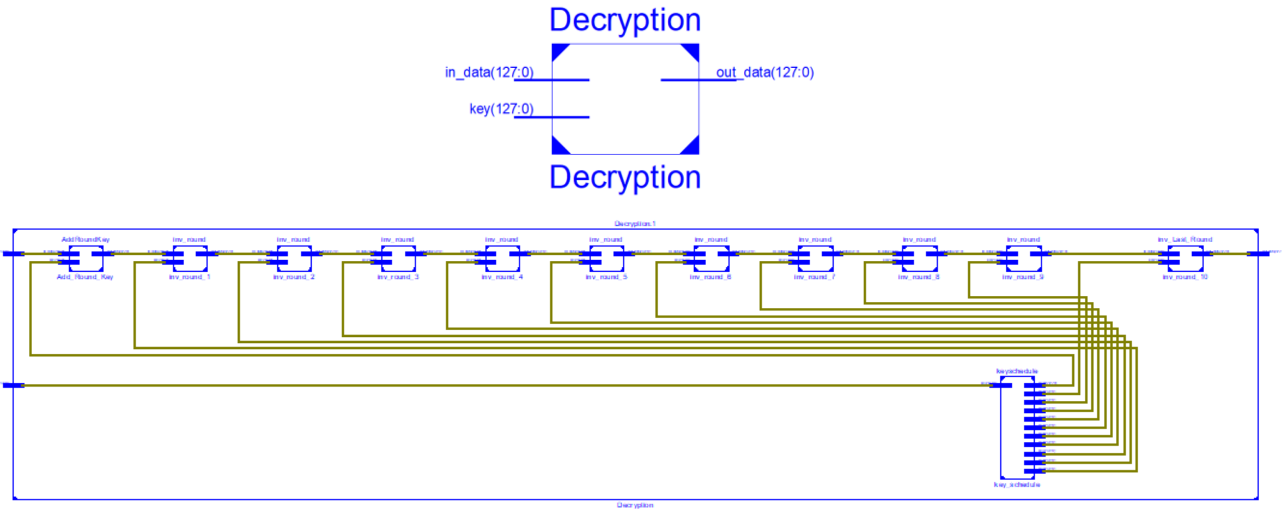
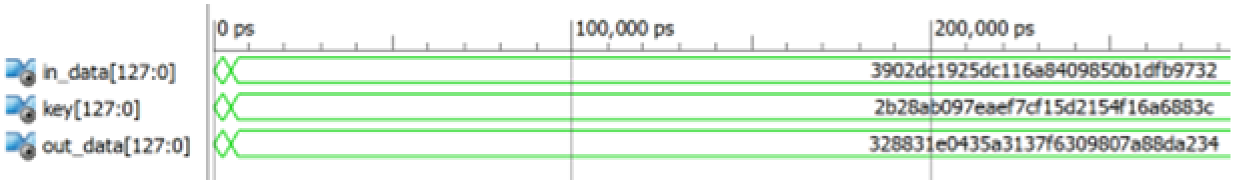


Figure3.16.1.RLT Schematic decryption

The RLT Schematic shows the 9 rounds and the last one with the key generator, which generates 10 keys each one will be used during its round. The distribution of the keys was well explained in Key Schedules process. The input of decryption process is the same of the first inv_round, the key input of decoding process will be the same of key schedule process. Key schedule will manage the keys from the tenth to the first for the inv_rounds with Reverse order. Each output of any inv_round will be the input of the following one,respecting Keyes’s management .the last inv_round’s output is the same of the decryptor.



```
in_data[127:0]    3902dc1925dc116a8409850b1dfb9732
key[127:0]        2b28ab097eaef7cf15d2154f16a6883c
out_data[127:0]   328831e0435a3137f6309807a88da234
```

Figure3.16.2.Simulation of decryption

10. Conclusion

We introduced this chapter by a brief description of the different tools necessary for the development of our crypto-system by presenting the different stages of implementation on a programmable circuit, and by describing the hardware description language used.

Subsequently, we have presented our results of simulations of each functional block to certify their proper functioning, by validating them with an example. At the end, we implemented our crypto-system on a Virtex-7 circuit. The obtained results are consistent for both encryption and decryption blocs.

Bibliography

[1] final-yearproject. [Online]. https://www.final-yearproject.com/2011/09/advanced-encryption-standard-cse.html.

[2] Zhihong and Cao, Lei and Su, Weilian and Wang, Tingkai and Yang, Huamin Qian, Recent advances in computer science and information engineering.: Springer Science & Business Media, 2012.

[3] Kenneth W and Lin, Herbert S Dam, Cryptography's Role in Securing the Information Society (Роль криптографии в защите информационного общества).: National Academy of Sciences, 1997.

[4] Czesław and Kurkowski ,Mirosław and Srebrny,Marian Koscielny, Modern Cryptography Primer.: Springer, 2013.

[5] Christof and Pelzl, Jan Paar, Understanding cryptography: a textbook for students and practitioners.: Springer Science & Business Media, 2009.

[6] Douglas Robert and Paterson, Maura Stinson, Cryptography: theory and practice.: CRC press, 2018.

[7] Mohammad and Boudriga, Noureddine Obaidat, Security of E-systems and Computer Networks.: Cambridge University Press, 2007.

[8] Gilbert Held, Network Design: Principles and Applications.: CRC Press, 2000.

[9] Hadi and Krutz, Ronald L Nahari, Web commerce security: design and development.: John Wiley & Sons, 2011.

[10] Jean-Philippe Aumasson, Serious cryptography: a practical introduction to modern encryption.: No Starch Press, 2017.

[11] Hamid R Nemati, Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering: Information Encryption and Cyphering.: IGI Global, 2010.

[12] Jahankhani and Gianluigi and Lilburn, Watson David and Frank, Leonhardt Hamid, Handbook of electronic security and digital forensics.: World Scientific, 2010.

[13] Keith M Martin, Everyday cryptography., 2012.

[14] Daemen and Vincent, Rijmen Joan, Information Security and Cryptography The design of Rijndael: AES-the advanced encryption standard.: springer, 2002.

[15] Minli Dai, Innovative Computing and Information: International Conference, ICCIC 2011, Wuhan, China, September 17-18, 2011. Proceedings.: Springer, 2011.

[16] Mano Paul, Official (ISC) 2 guide to the CSSLP CBK.: CRC Press, 2013.

[17] Kazuo and Sasaki, Yu and Li, Yang Sakiyama, Security of block ciphers: from algorithm design to hardware implementation.: John Wiley & Sons, 2016.

[18] Jonathan, and Yehuda Lindell Katz, Introduction to modern cryptography.: CRC press, 2014.

[19] Tom St Denis, Cryptography for developers.: Elsevier, 2006.

[20] Hans, Helmut Knebl, and Helmut Knebl Delfs, Introduction to cryptography. Vol. 2.: Springer, 2002.

[21] R., Jejurkar, R., Chopade, S., Vaidya, S., & Sanap, M. Jain, "AES algorithm using 512 bit key implementation for secure communication.," *Int. J. Innov. Res. Comput. Commun. Eng*, 2014.

[22] Nidhal and Guilley, Sylvain and Danger, Jean-Luc Selmane, *Practical setup time violation attacks on AES 2008 Seventh European Dependable Computing Conference*.: IEEE, 2008.

[23] Adam Berent. (2013, June) networkdls. [Online]. http://www.networkdls.com/Articles/AESbyExample.pdf

[24] Uli Kretzschmar, "Aes128--ac implementation for encryption and decryption," *TI-White Paper*, 2009.

[25] Hyubgun and Lee, Kyounghwa and Shin, Yongtae Lee, "Aes implementation and performance evaluation on 8-bit microcontrollers," *arXiv preprint arXiv:0911.0482*, 2009.

[26] Roshni and Patel, Aamna Padate, "Encryption and decryption of text using AES algorithm," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, pp. 54--9, 2014.

[27] Jay Dave, and Erez Zadok Charles P. Wright, "Cryptographic File Systems Performance: What You Don’t Know Can," *Appears in the proceedings of the 2003 IEEE Security In Storage Workshop (SISW 2003)*, vol. 17, 2003.

[28] Sanjay and Hyderabad, I Churiwala, *Designing with Xilinx FPGAs*.: Springer, 2017.

Bibliography

[29] Ian and Tessier, Russell and Rose, Jonathan Kuon, *FPGA architecture: Survey and challenges*.: Now Publishers Inc, 2008.

[30] Umer and Marrakchi, Zied and Mehrez, Habib Farooq, *Tree-based heterogeneous FPGA architectures: application specific exploration and optimization*.: Springer Science & Business Media, 2012.

[31] DR.Hamza.ATOUI. (2020, May) researchgate. [Online]. https://www.researchgate.net/publication/341406833_Rappel_sur_le_langage_VHDL_pour_les_masters_II_SEM

[32] Jean-Michel and Fonkoua, Alain and Maginot, Serge and Rouillard, Jacques Berge, *VHDL designer's reference*.: Springer Science \& Business Media, 2012.

[33] Ronald and Schmidt, Andrew G Sass, *Embedded systems design with platform FPGAs: principles and practices*.: Morgan Kaufmann, 2010.

[34] Len Gelman, *Electronic engineering and computing technology*.: Springer Science & Business Media, 2010.

[35] Charles H and John, Lizy K Roth Jr, *Digital systems design using VHDL*.: Nelson Education, 2016.

[36] openclassrooms. [Online]. https://openclassrooms.com/fr/courses/3028701-protegez-l-ensemble-de-vos-donnees-sur-votre-ordinateur/3110141-le-chiffrement-des-donnees-qu-est-ce-que-c-est

[37] Jonathan and Lindell, Yehuda Katz, *Introduction to modern cryptography*.: CRC press, 2014.

[38] Jean-Philippe Aumasson, *Serious cryptography: a practical introduction to modern encryption*.: No Starch Press, 2017.

[39] Network Security for Two End Users. [Online]. https://ukdiss.com/examples/network-security-for-two-end-users.php?vref=1

[40] Jahankhani and Gianluigi, Me and Lilburn, Watson David and Frank, Leonhardt Hamid, *Handbook of electronic security and digital forensics*.: World Scientific, 2010.

[41] finalyearthesis. [Online]. http://www.finalyearthesis.com/

Bibliography

[42] Ryan Silva. (2007, March) researchgate. [Online]. https://www.researchgate.net/publication/235039657_Implementation_and_Optimization_of_the_Advanced_Encryption_Standard_Algorithm_on_all_8-Bit_Field_Programmable_Gate_Array_Hardware_Platform

[43] Samir El Adib Ismail Negabi. (2019, 25 Sep) hal.archives-ouvertes. [Online]. https://hal.archives-ouvertes.fr/hal-02296924/document

[44] Athanasios Kakarountas,Apostolos Fournaris,Athanasios Milidonis,Odysseas G. Koufopavlou Georgios Selimis. (2007, December) researchgate. [Online]. https://www.researchgate.net/publication/220091765_A_Low_Power_Design_for_Sbox_Cryptographic_Primitive_of_Advanced_Encryption_Standard_for_Mobile_End-Users

[45] Nada Hussein M. Ali Abdul Monem S. Rahma. (2015, April) researchgate. [Online]. https://www.researchgate.net/publication/312277403_An_Improved_AES_Encryption_of_Audio_Wave_Files

[46] Apostolos Fournaris,Odysseas G. Koufopavlou Georgios Selimis. (2007, January) researchgate. [Online]. https://www.researchgate.net/publication/224715599_Applying_Low_Power_Techniques_in_AES_MixColumnInvMixColumn_Transformations

[47] brainkart. [Online]. http://www.brainkart.com/article/AES-Key-Expansion_8410/

[48] poisonninja. [Online]. https://poisonninja.github.io/2016/12/07/AES-in-Google-Sheets/

[49] Ayushi and Malhotra, Mohinder Arya, "Effective AES Implementation," *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, vol. 7, no. 2016.

[50] S and Prayline Rajabai, C Ramanathan, *FPGA Implementation of an Area Optimized Architecture for 128 bit AES Algorithm*.

[51] formaestudio. [Online]. http://www.formaestudio.com/rijndaelinspector/

General

Conclusion

General Conclusion

New design technologies for electronic systems require a high level of abstraction and a very structured methodology to allow the designer to devote himself solely to design without worrying about the details of implementation at the schematic and integration level.

The advantages of hardware implementation are, by nature, more secure physically, faster (broadband), more reliable. FPGA circuits are programmable logic circuits whose implemented function is not fixed, and they can be programmed several times on site. It is therefore a promising alternative for implementing block encryption.

In this work, we presented a hardware implementation of Rijndael's 128-bit AES (Advanced Encryption Standard) encryption and decryption algorithm on a reconfigurable platform based on FPGA (Field Programmable Gâte Arrays).

Our study led us to consider the following points:

- First, we started with a quick overview in the field of encryption and the existing encryption algorithms.
- Then we focused on AES algorithm by its theoretical aspects. We studied the mathematical operations required by the AES rounds namely :SubBytes, ShiftRows, MixColumn and AddRoundKey before exposing the proposed pipeline approach for the algorithm.
- In the second part we presented our proposed architecture for Encryption (and Decryption) system. The architecture of the system has been described in VHDL language using ISE Design Suite of Xilinx. The developed IPs have been tested and simulated with the ISim simulator of ISE Design Suite.
- Our design has been simulated and validated, and the obtained results have been verified. All the various modules are fully functional which makes our system operational.

As a perspective to this work, an optimized pipeline architecture will further increase the efficiency of the system. As well as the use of our architecture on real data, with hardware implementation on rapid prototyping target, can materialize the efficiency of our proposed.