ENCRYPTION OF SATELLITE IMAGES WITH AES ALGORITHM ON APACHE SPARK

2020
MASTER THESIS
COMPUTER ENGINEERING

Muhammad Yaseen ALHAYANI

Assist.Dr. Yasin ORTAKCI

ENCRYPTION OF SATELLITE IMAGES WITH AES ALGORITHM ON APACHE SPARK

Muhammad Yaseen ALHAYANI

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
Master Thesis

Assist.Prof.Dr. Yasin ORTAKCI

KARABUK
January 2021

I certify that in my opinion the thesis submitted by Muhammad Yaseen ALHAYANI titled “ENCRYPTION OF SATELLITE IMAGES WITH AES ALGORITHM ON APACHE SPARK ” is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist.Prof. Dr Yasin ORTAKCI .....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. Jan 18, 2021

Examining Committee Members (Institutions)	Signature
Chairman : Assist.Prof.Dr. Abdullah ELEN (BOEU)	.....
Member : Assist.Prof.Dr. Emrullah SONUÇ (KBU)	.....
Member : Assist.Prof.Dr. Yasin ORTAKCI (KBU)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Hasan SOLMAZ .....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Muhammad Yaseen ALHAYANI

ABSTRACT

M. Sc. Thesis

ENCRYPTION OF SATELLITE IMAGES WITH AES ALGORITHM ON APACHE SPARK

Muhammad Yaseen ALHAYANI

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Asisst. Prof.Dr. Yasin ORTAKCI
January 2021, 68 pages

In this thesis, we aim to protect the privacy of huge satellite images chunks by encrypting them with the AES algorithm in three stages. The first stage is to encrypt it by Python programming language and calculate the elapsed time for the encryption and decryption process. In the second stage, we implemented the algorithm inside Spark's environment, and calculating the elapsed time for the encryption process. In the third stage, we implemented the encryption process inside the multi-node cloud, and we calculated the elapsed time for the execution. Through the results, the third stage was the fastest in encrypting and decrypting satellite images based on the cloud. Also, we measured the AES encryption ability inside the cloud by scale up and speed up creteria. We concluded that the ability of the AES algorithm to gain time is high, and the system is balanced because the expansion of the data volume does not affect the encryption result. In the end, we concluded that encrypting satellite images on the cloud improved the performance of the encryption package. More nodes and

specifications of the CPU for the cloud increased the speed of the algorithm and their abilities. The execution time of an algorithm on Python was 880.33 sec for three thousand images. For cloud, the execution time decreased to 54.12 sec for the same number of images, which increased the ecryption speed 40% more than standalone mode.

Key Words : Apache Spark, Big data, AES algorithm, Satellite images, cloud computing.

Science Code : 92414

ÖZET

Yüksek Lisans Tezi

APACHE SPARK ÜZERİNDE AES ALGORİTMİ İLE UYDU GÖRÜNTÜLERİNİN ŞİFRELENMESİ

Muhammad Yaseen ALHAYANI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Yasin ORTAKCI
Ocak 2021, 68 sayfa

Bu tezde dev uydu görüntülerini üç aşamada AES algoritması ile şifreleyerek gizliliğini korumayı hedefliyoruz. İlk aşama, Python programlama dili ile şifrelemek işlemini gerçekleştirdik ve hem şifreleme hem de şifre çözme işlemi için geçen süreyi hesapladık. İkinci aşamada, algoritmayı Apache Spark ortamına uyarladık ve şifreleme işlemi için geçen süreyi hesapladık. Üçüncü aşamada şifreleme sürecini çok düğümlü bir bulut kümesine uyguladık ve yürütme için geçen süreyi hesapladık. Sonuçlar Spark ortamında gerçekleştirilen bulut tabanlı uydu görüntülerini şifreleme ve şifresini çözme işleminin daha hızlı olduğunu göstermektedir. Ayrıca, bulut içindeki AES şifreleme yeteneğini ölçmek için hızlandırma ve ölçekleme kriterlerini kullandık. AES algoritmasının hızlandırma yeteneğinin yüksek olduğu ve veri hacminin genişlemesinin şifreleme sonucunu etkilemediği sonucunu elde ettik. Sonunda, uydu görüntülerini bulutta şifrelemenin şifreleme paketinin performansını iyileştirdiği sonucuna vardık. Bulut için daha fazla düğüm kullanılması algoritmanın hızını ve

yeteneklerini artırdı. Python'da bu algoritmanın çalıştırılma süresi 3000 uydu görüntü için 880.33 saniyeydi. Bulut için, aynı sayıda görüntü için yürütme süresi 54,12 saniyeye düştü. Bu sonuç şifreleme hızının %40 arttığını göstermektedir.

Anahtar Kelimeler : Apache Spark, Büyük Veri, AES algoritması, Uydu Görüntüleri, Bulut Bilişim.

Bilim Kodu **:** 92414

ACKNOWLEDGMENT

First, I thank God for my success, also i would like to thank my supervisor, Dr. Yasin ORTAKCI, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I also thank Dr. Sohaib K.M. Abujayyab who supported me and helped me to get the research thesis data set.

Besides, i would like to thank my parents for their wise counsel and sympathetic ear. You are always there for me; I could not have completed this dissertation without the support of you.

I also thank my dear friends Sipal and Salma who were stood with me, and a special thanks to my friend Ayman Elbelasy, who was my right hand at work.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVITIONS INDEX

ABBREVITIONS

DAG : Directed Acyclic Graph

RDD : Resilient Distributed Dataset

NIST : National Institute of Standards and Technology

IDA : Iterative Deepening Algorithm

ECDH : Elliptic-Curve Diffie–Hellman

EC : Elliptic-Curve Cryptography

RSA : Rivest–Shamir–Adleman

RC6 : Rivest cipher

DES : Data Encryption Standard

TFTP : Trivial File Transfer Protocol

AES : Advanced Encryption Standard

PART 1

INTRODUCTION

Data privacy and protection in this era has become the first need that is focused on. The large increase in the volume of data, and its entry into all areas of life make its protection necessary. An actual example of a data collection medium is multimedia, also logistical and financial databases, social networks, sensors, the internet of things, etc. The privacy and security of this data is the basic factor in all these media. The privacy achieved by safely storing and sharing big data [1].

Images are one of the most important and sensitive sources of stored data, because the data in a small image can be equivalent to many pages of text data. Consequently, images must be secured to prevent unauthorized users from accessing the content of the images, during storage or transfer it by transport media [2]. One of the important topics today in data protection is securing satellite images, because it is important sources in the field of various sciences. Uses of satellite imagery; the field of national security, military operations, weather forecasting, monitoring of land resources, tectonic activity, surface plants, geological research and training [3].

After these images are taken from the monitoring satellites of the earth, it is unsafe to transmit them to the ground stations through communication channels without any opportunity to be vulnerable to penetration [4]. We used the encryption method with images to protect them from unauthorized people. Even if images obtained through hacking operations they can not get the image content [5]. We will use Advanced Encryption Standard (AES) algorithm in the image encryption process. AES is a type of symmetric encryption system that uses one key for the encryption and decryption process. Symmetric encryption is the most suitable for encrypting large amounts of data. the AES named Rijndael algorithm is the best in security, performance, efficiency, implementation capacity and flexibility. Depending on research conducted

by the NIST, Rijndael's algorithm was better than all the algorithms suggested in the safety factor [6].

In this thesis, we applied the AES algorithm to encrypt and decrypt satellite images, and we measure the encryption performance of AES algorithm with Apache Spark in cloud environment. Spark has many advantages in processing large amounts of data and has an advanced DAG engine that supports periodic data streaming. Also, Spark with its in-memory computing shows better performance than Hadoop in terms of speed [7].

We encrypt the satellite images with the AES algorithm, then the elapsed time for the encrypt and decrypt process will be calculated. After that, we adapting AES algorithm to the Apache Spark environment. Also, we are going to encrypt different sizes of satellite imagery inside Spark and make performance comparisons by calculating the elapsed time. We applied AES inside clusters with several nodes in the Spark environment [8].

Figure 1.1. Architecture of encryption operation.

We analyze the results of huge satellite image chunks encryption ability with scale up[9], Speed up [9] criteria. These measures are used to evaluate the performance of the AES algorithm, and we measure algorithm speed in the encrypt and decrypt process. All work will be performed in Python [9].

1.1. PROBLEM STATEMENT

Companies are increasingly relying on analysis using big data in different areas. Thus, sensitivities of big data security are an obstacle that organizations must overcome. When big data was encrypted with encryption techniques to secure it, it was discovered that it took a long time. It needs to increase resources to deal with time problems and may miss critical deadlines for customers, which lead to revenue and business loss. We're going to use the Apache Spark environment in the cloud, to develop encryption technology in order to get as quickly as possible time to encrypt big data.

1.2. THE CONTRIBUTIONS

In this thesis, AES algorithm is adapted to the Apache Spark environment based on cloud. We performed the initial encryption process for a group of images, and we calculated the elapsed time for that process. We discovered faster results than using the AES algorithm alone. After the success of encryption within cloud, it can be used to build and develop many security applications in the future to secure big data. After that, we developed the encryption process and increased its efficiency by applying it inside the cluster for multiple nodes. Approximately 3,000 satellite images are collected. As more nodes and more cluster specifications of CPU increased, more time was gained, and the coding time decreased to less than a minute at 16 nodes. For this, the speed factor was achieved in real-time within the encryption process. We developed AES ability and adapt it within the Apache Spark environment to accommodate the encryption of huge satellite images.

1.3. AIM OF THE STUDY

The main purpose of this study is to succeed in encrypting satellite images through the AES algorithm in the shortest possible time. We aim to improve AES ability by Apache Spark and to protect satellite images from hacking attacks, and unauthorized access with encryption and decryption process in the fastest time.

1.4. DISSERTATION OVERVIEW

This dissertation is organized as follows. Chapter1, a general background is provided on the thesis. Also, the contribution and aim of the thesis is described.

Chapter2 covers what the researchers suggested of techniques for encoding satellite images close to our work. The conclusions of other studies and comparing them with our work.

Chapter3 introduces the AES algorithm and the steps used in the encryption and decryption process. Also, we compared it with the DES algorithm.

Chapter4 explains the Apache Spark environment, its architecture and components. Along with RDD technology and adaptation to the AES algorithm on cloud computing.

In chapter5 the time gain features are discussed. Then, how to implement the algorithm inside Spark and within the cloud. Also, showing results.

Chapters6 presents the experimental results with the graphics, as well as discussion of these results and their arrival to the goal.

Chapters7 covers the conclusions based on those results and what the future work plans are.

PART 2

RELATED WORK

In this section, we aim to highlight work done by others that somehow ties in with our own work. Works include working methods, the various techniques used to encrypt and decrypt satellite images. Also, how we inspired to find this topic through another research.

Canabay et. al focused on the privacy and protection of big data, and Apache Spark was proposed [10]. The protection technology used with the Spark environment was data masking. The anonymity of big data has been successful and faster with Spark. The privacy of big data has been protected by hiding it through the proposed model. For our research, the privacy of data for satellite images protected by encryption technology instead of masking technology. With the Apache Spark environment, after encrypt the satellite image data, we achieved higher security on the data. We ensure that if the hacker obtained the data, and was able to display it, he can not access the data content because it's encrypted.

Shah et. al presented a study on the RDD data structures inside Apache Spark [11]. They explained that RDD stores and distributes data in memory without protection. Data is vulnerable to penetration in the event of storing huge data. They discovered that, Apache Spark is not suitable for processing sensitive information. They created a solution by combining IDA algorithm and Shamir's perfect secret sharing (PSS), to provide a strong security system for secure data inside the memory. In our study, when dealing with Apache Spark, no issue for securing the storage of distributed data inside the memory, because the data is encrypted by AES algorithm before storing it in the memory. Even if data stored in cloud computing, the process of data encryption is before the operation storage.

Bhargavi et. al conducted a study in securing big data, through cloud computing using Hadoop with a set of symmetric and asymmetric encryption algorithms [1]. After success in encrypting the data, they found a large discrepancy in the results of implementation time between algorithms. Algorithms with good results are ECDH, then EC and AES. The execution time results for the rest of the algorithms were less rapid. In our research, the big data for satellite images were secured with the symmetric encryption algorithm AES. The work was inside the Apache Spark environment not with Hadoop, because it is faster than Hadoop in the distribution and storage of data. After all, Spark has RDDs data structures that distribute the data. Data distribution made the AES algorithm faster in encryption and decryption than other algorithms.

In another study [12], Al Mamun worked on securing data stored inside the cloud by using the RSA asymmetric encryption algorithm. RSA algorithm works on encrypting small-sized data. They created a model called big crypt, that allows encryption techniques to work on big data using symmetric and asymmetric key. The data successfully secured, and the model with this algorithm proved successful in encrypting large data. In our research, to protect the data stored in the cloud, we used AES symmetric encryption algorithm. The AES algorithm has one key that is used in the encryption and decryption process, unlike the RSA algorithm which has a private and a public key for the encryption and decryption process. We concluded that AES is faster than the RSA algorithm because it hase one key for encryption. Also, AES algorithm is used to encrypt big data without the need for any form.

In [13], Chen et. al have planned to improve the encrypting speed of AES algorithm on big data, by relying on deep pipelines and expanding the technology completely. This has been successful and has achieved good transmission speed. As for our work, the encryption has been improved for the AES algorithm using the Apache Spark with multi-node cluster. The results showed high encryption speed with real execution time.

In [7] and [14] different methods were developed for encoding video, depending on the Hadoop MapReduce and Apache Spark environment. They used AES algorithm for encryption. The video was cut into small video units, and distributed these units through distribution environments and send it to server nodes for encryption. The

encryption is done through one node and a group of distributed nodes. It has been proven that the encoding in the distributed video process is more efficient with Spark and its encryption speed was higher than Hadoop. For our work, we downloaded a large number of satellite images, and we captured it via SAS planets program instead of video. In the process of distributing the data, Hadoop MapReduce was not used because it is slower. Instead, RDD was used inside Spark to distribute the image data to the nodes in memory. In the encryption process inside the cloud, the CPU specifications were increased and node numbers were increased to more than 16. The encryption was executed many times on the images. Thus, we reduced the encryption process time greatly to get as little as possible.

In [15], Mehmood stated that internet of things devices can generate huge data in their work. They used cloud computing to store the data. The main problem is needed to secure this data stored in the cloud. He used the encryption technology to preserve the privacy of IoT data stored inside the cloud. algorithms used were AES and DES. It found that these algorithms are complex and difficult to deal with to encrypt IoT devices data. They created a lightweight encryption system for the cloud. They studied the complexity of the proposed algorithms, and compared them to use an appropriate encryption on IoT data stored in the cloud. This study expanded more encryption techniques and to know their complexity and application areas. In our study, we are dealing with satellite images stored within cloud computing. Its compatible with AES algorithm, and without any complexity. The data of these images are converted into bytes to be easily readable within the algorithm block. The data encrypted without the need to form a model lightweight encryption or measure the complexity of an algorithm. The encrypted images stored in the cloud.

In [16], Aljawarneh et. al studied multimedia. Multimedia data is one of the most important big data, because of the magnitude and complexity of this data. Symmetric encryption technology was used for a group of algorithms such as DES, RC6, and AES to encode this data, to secure it from any hacking process. The researchers thought that these algorithms may be slow in the encryption process on the multimedia data. They developed a cipher algorithm to speed up the encryption process on this data. It works with a multi-indicator programming system. This system encrypts multi-level

encryption for different encryption algorithms, such as the feistel coding system and the genetic algorithm with AES. After testing and comparing its results with the results of the traditional algorithms. The system showed less implementation time and higher efficiency than other algorithms. In our research to encode satellite images, the ability of the AES algorithm also had to be improved. We need to accommodate the huge volume of images and encode them in the shortest possible time. There is no need to build a new multi-path coding system. We improved the performance of the algorithm itself, and developed its capacity and speed by adding it to the Apache Spark environment with cloud computing in a multi-node. Thus, we have succeeded in improving the performance of the algorithm, and increasing its speed of implementation.

In [17], Velan et. al have focused on a step beyond data encryption, which is the step of encrypted traffic and data transmission across the network. They proposed some methods for measuring encrypted traffic, and methods that depend on the type of traffic while studying encrypted traffic protocols. In our study, we deal with cloud computing in storing data after it is encrypted. Encrypted images is stored in the cloud computing, when the transfer and data sharing process started, as mentioned here [18], the cloud computing resources and its power centers help in the collective security for the transfer process data through secure traffic.

In [19], Gai and others interested in the privacy of big data. They created a dynamic encryption model called D2ES supported by the DED encryption algorithm. They use algorithms to encode big data by data encryption selection system, and privacy classification to increase its level. They applicated the work in the cloud computing. The results showed progress and different implementation times. In the end, the proposed approach proved its efficiency in strengthening the privacy of big data. In our study, the proposed approach was in three stages to reach the highest level of data privacy. In the first level, we encrypted the satellite images with the AES algorithm by Python. In the second level, we adopted and implemented the algorithm within the Apache Spark environment to increase the speed of the algorithm. In the third level, the algorithm was implemented by Spark within the cloud. The third level has achieved its efficiency at the rest of the levels, and the work of the algorithm has been greatly

improved. Also, the speed of implementation has increased to encrypt and decrypt the satellite images.

In [20], Sekar et. al interested in the encryption of big data before storing it to the cloud computing. They used Hadoop environment to process and interpret the results of big data, by dealing with symmetric and asymmetric encryption algorithms such as AES, DES, and RSA. They presented an integrated approach to encrypting and decrypting big data, to achieve better performance results of encryption operations. They conclude that the AES algorithm has less time in implementing the encryption and decryption process. Also, consumes less space in storage than the rest of the algorithms. In our study, we did the process of encoding the big data for satellite images inside the cloud. We used AES algorithm with the Apache Spark environment, because it is faster than Hadoop. the performance of the algorithm improved more. Also, the execution time of the encryption process was obtained in the real time.

In [21], Zhang et. al interested in encrypting digital images through the AES algorithm. The algorithm has been implemented in Matlab language then, implement the encryption process on digital images. After the results appeared, the researchers discovered that this process has a good effect with the encrypted images which have been decrypted. The algorithm has achieved good success and high efficiency. In our study, attention was paid to encoding satellite images through the AES algorithm. This algorithm was represented in the Python language because it supports symmetric encryption libraries. After implelment the algorithm inside Python, the satellite image data is entered into the algorithm in a byte, then the encryption process has been conducted. After displaying the results and the the execution time, we compared the execution time with other algorithms such as DES, we found AES faster and more efficient.

In [22], Bensikaddour et. al discussed how to protect the satellite images transferred from the monitoring satellites to the earth station. They proposed encryption technology to protect the privacy of the transmitted images. The proposed encryption technology is a multi-spectrum encryption process based on the Fridrich's scheme. After success of the encryption and decryption process on the images, the results

proved that this method has a level of good security, but with low complexity on the encrypted images. Also, the algorithm has low encryption strength and speed as well as energy consumption. For our research, we dealt with the process of encoding satellite images by AES algorithm. Upon reviewing the results, we found that this method has a very high-security level. Also, the level of complexity of the images is high with the good encryption speed. Also, the stored images are small due to the encryption speed.

In [23], Xu et. al interested in encrypting big data using the distributed encryption system through Hadoop. They discovered that encrypting big data with the traditional algorithms is not fast and not applicable. They proposed the RSA algorithm and improved it by integrating it with MapReduce in Hadoop. So, the encryption process is distributed on the proposed algorithm. After encryption, the encrypted text is merged into one part. After show the results, they discovered that RSA improved the speed of the encryption process and it could be used in encrypting big data. In our study, we encrypted the big data of satellite images by AES algorithm with a distributed encryption system. We used Apache Spark instead of Hadoop because it is faster than it. By using the RDD to distribute data and store it inside the memory instead of the hard disk as Hadoop does. The results showed the effectiveness of this technology with its high speed in encrypting images.

In [24], Reddy et. al focused on protecting the images during their transfer by encrypting them, to prevent unauthorized persons from accessing to their contents. They encrypting the images using the AES algorithm after its development. The concept of changing the keys (EKE) is used with the AES algorithm, which allows the secret key exchange between both the client and the server. The data transfer takes place by means of the TFTP protocol, through which the data is transferred between the server and the client securely. A graphical user interface for the encryption and decryption of images has been implemented. All details are implemented in Linux environment using the OpenCV- Python script. From experimental results, the improved AES proved to be more efficient in encryption, and in saving time compared to other algorithms such as DES3. In our study, AES was used to encrypt satellite images, and prevent unauthorized persons from accessing the identity of the data. As

for the secret key, it is secretly created on the server, and the client receives a copy of it before the transfer and encryption process. Since the data is stored in the cloud, depending on that. The cloud provides a medium for sharing data securely. A graphical user interface has also been implemented to encrypt and decrypt all kinds of data, such as images, videos, and files, all details have been implemented in a Windows environment using Crypto Cipher- Python script. In the end, the operation proved successful getting encrypted data with high speed and real execution time.

2.1. SOURCES OF THESIS IDEA

In the field of image encryption, many researchers have studied image encryption techniques, as [25], where Mahalakshmi et. al concluded the traditional encryption techniques are insufficient to secure images. They said it is advisable to rely on algorithms and merge them with traditional methods. Thus, they came up with safer and more effective ways to ensure the privacy of images, this drew attention to the factor of integrating technologies with cryptographic algorithms. In the field of data protection, researchers have developed techniques to conceal the identity of big data and have been tried on both Hadoop and Spark. It was discovered that Spark is faster than Hadoop. Spark solves the problem of data flow in real-time. In conclusion, a model based on Spark was proposed for the first time to protect data. Spark first technology that facilitates the process of developing big data protection [10].

We got a good example to use Spark with the proposed encryption algorithm instead of other technologies. Subramanyan have studied the ability of Rijndael's algorithm and applied some methods and operations on it in the field of encoding images. He found that this algorithm has a high capacity to resist hacking attacks, key sensitivity tests, and statistical coding analysis [26].

These promising results are related to our current study by securing big data, through the application of technologies within cloud computing. Researchers began to apply D2ES techniques, to improve the efficiency of data privacy with the DED algorithm for encrypting data within the cloud. The experimental results have proven the success and superiority of the proposed approach. Gai takes the opportunity to encrypt data

inside the cloud [19]. We tried to get a good and short execution time, by increasing the number of nodes within the cloud on Spark to gain more time [27].

In the field of big data security. Companies suffer from the difficulty of maintaining and sharing big data safely, and not ensuring its security and vulnerability to penetration. Thus, special techniques must be relied upon to secure this data. Concerning to [28], Terzi et. al have studied techniques for securing data. They confirmed that, the big data can be secure when encrypted with appropriate standards, it's inspired us to use the encryption method for protecting data.

2.2. COMPARISON THE ALGORITHMS

For data protection, researchers made a comparison between three encryption techniques AES, DES, and RSA based on an analysis of the execution time [26, 29]. After the encryption and decryption operations that were performed on different data, the elapsed time was calculated for each algorithm. As shown in the figures (2.1, 2.2), AES algorithm took less time from the rest of the algorithms in the encryption and decryption process [26]. AES algorithm can be relied to demonstrate its efficiency in the speed factor over other algorithms [29].

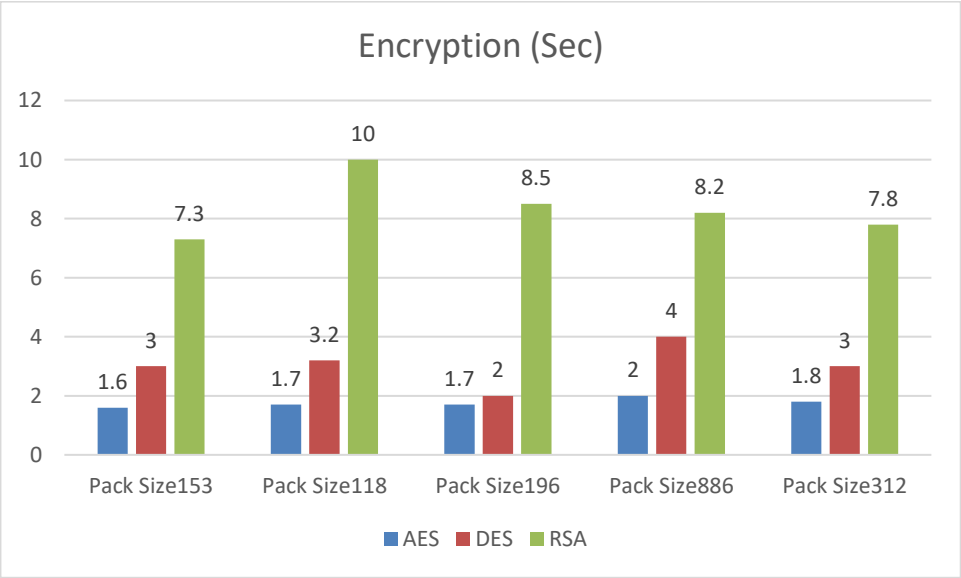

Figure 2.1. Encryption operation[30].

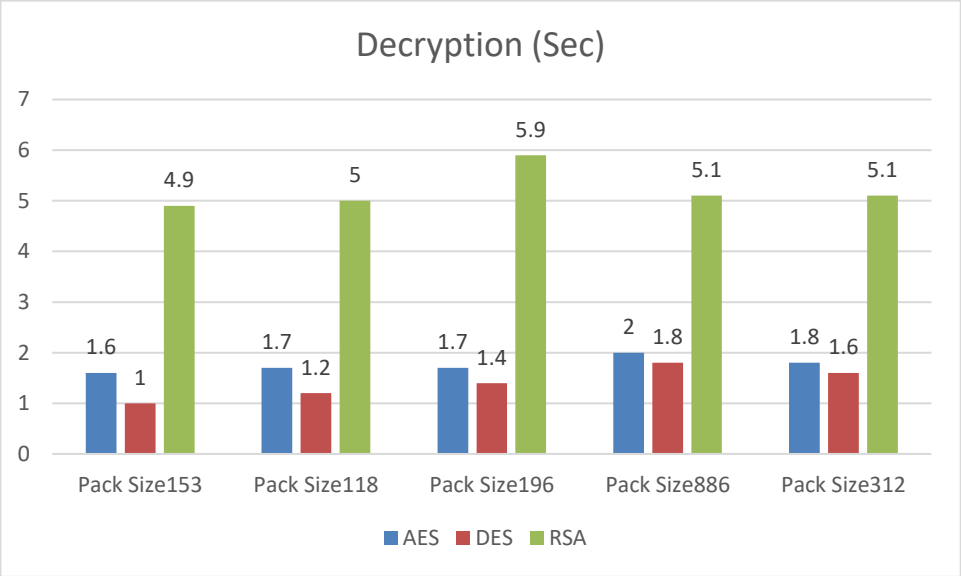

Figure 2.2. Decryption operation [30].

In another study [31], Alanazi et. al conducted a comparison process on data encryption algorithms, including AES, DES and multiple DES in terms of efficiency, flexibility, speed of performance with key length. based on nine different factors as in table 2.1, it shows the superiority of AES over the rest of the algorithms.

2.3. ADAPTATION TO TECHNIQUES

The algorithm must be ensured that it works with other techniques and coding systems. Aljawarneh and others developed a multi-level encryption system to encode big data using the AES algorithm. The results were that the encryption process had less running time with increased productivity of encryption, compared to traditional encryption algorithms. The system results were good with the algorithm in terms of performance and safety. AES algorithm has highly efficient in integration with other encryption techniques, this help in adapting it to the Apache Spark platform[16].

Table 2.1. Comparison between AES, DES and 3DES [31].

Factors	AES	3DES	DES
Key Length	128, 192, or 256 bits	(kl, k2 and k3) 168 bits (kl and k2 is same) 112bits	56 bits
Cipher Type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher
Block Size	128, 192, or 256 bits	64bits	64 bits
Developed	2000	1978	1977
Cryptanalysis resistance	Strong against differential, truncated differential, linear, interpolation and square attacks	Vulnerable to differential, Brute Force attacker could be analyzing plaint text using differential cryptanalysis.	Vulnerable to differential and linear cryptanalysis; weak substitution tables
Security	Considered secure	one only weak which is Exit in DES.	Proven inadequate
Possible Keys	2123221922 or 22"	2 112 or 2163	2"
Possible ASCII printable character keys	9516, 952', or 9532	951' or 9521	95'
Time required to check all possible keys at 50 billion keys per second**	For a 128-bit key: 5 x 1021 years	For a 112-bit key: 800 Days	For a 56-bit key: 400 Days

PART 3

ENCRYPTION ALGORITHMS

For data encryption, there are two main encryption modern techniques, symmetric and asymmetric encryption, as shown Figure 3.1:

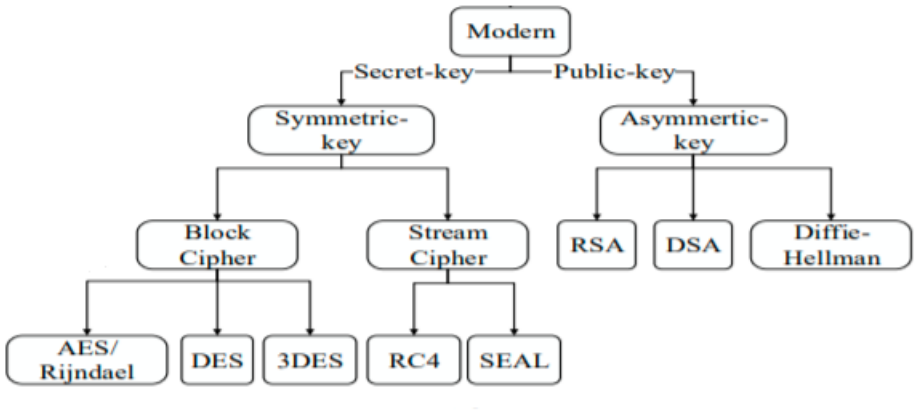

Figure 3.1. Classification of encryption technologies [32].

In [33], a comparison of these two techniques is performed. They concluded that the symmetric encryption technique is faster, because the sender needs to perform one encryption for all mixed servers. Asymmetric encryptioni slow and takes longer due to repeated encryption operations. Also, symmetric encryption techniques have one secret key for the encryption and decryption process. In contrast, asymmetric encryption techniques have two keys as private and public key for the encryption and decryption process.

3.1. ADVANCED ENCRYPTION STANDARD

AES algorithm, is one of the well-known and popular algorithms in symmetric encryption technology. It is also known as Rijndael, and It's proved effective on a large scale around the world. It was established by National Institute of Standards and

technology (NIST) in 2001. In [34], Abdullah mentioned that this algorithm deals with big data. AES algorithm has one key for encryption and decryption of three different sizes (128,192,256) bit with 128-bit block size. These 128 bits are represented in 4x4 array and AES operates on an array of bytes. AES has the feature of rounds that repeats the encryption process, to increase the complexity of the ciphertext. It also specifies the key sizes as AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys [35], as shown in Figure 3.2. At the end of the study, it was compared AES with other algorithms, such as 3DES, and proved its superiority over it in terms of speed for satellite images.

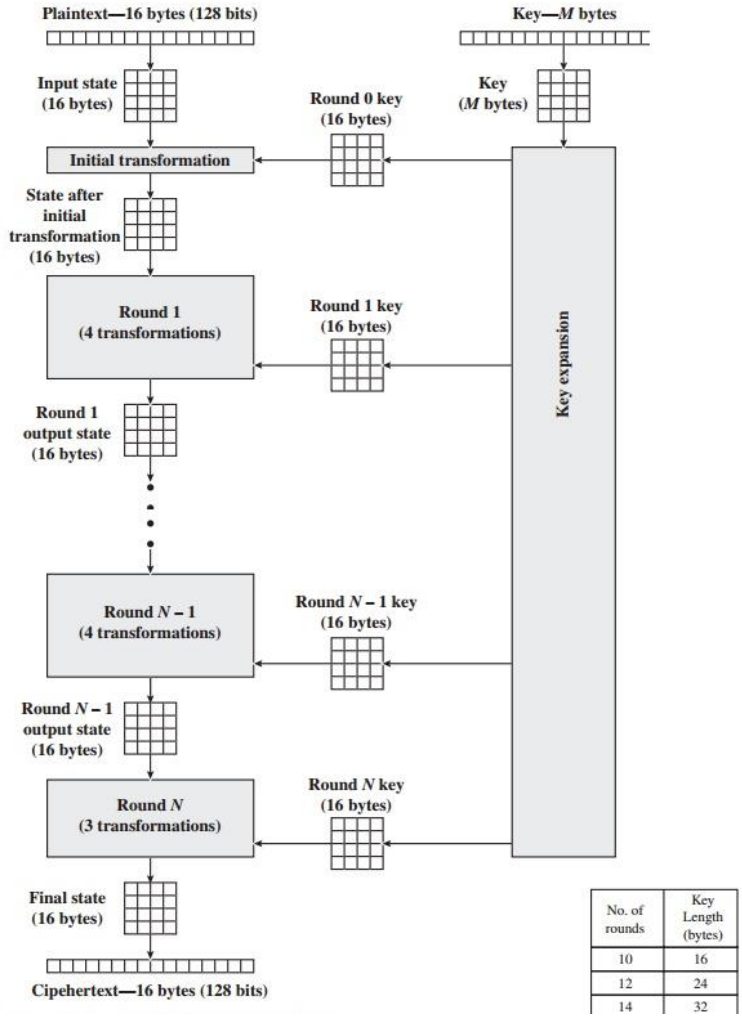

Figure 3.3. AES operations [35].

3.1.1. AES Architecture

Briefly, the algorithm uses symmetric block cipher and contains four transformations (SubBytes, ShiftRows, MixColumns, and AddRoundKey) in each round. When encrypting satellite images, their pixel values must be converted into bytes. After that, it entered into the state block, and the transform operations are performed on it. Encryption with these transforms is done after the key is generated. To perform the decryption process, the number of rounds depends on the length of the key, and the mentioned operations are reversed [36].The transformations of AES shown in Figure 3.3.

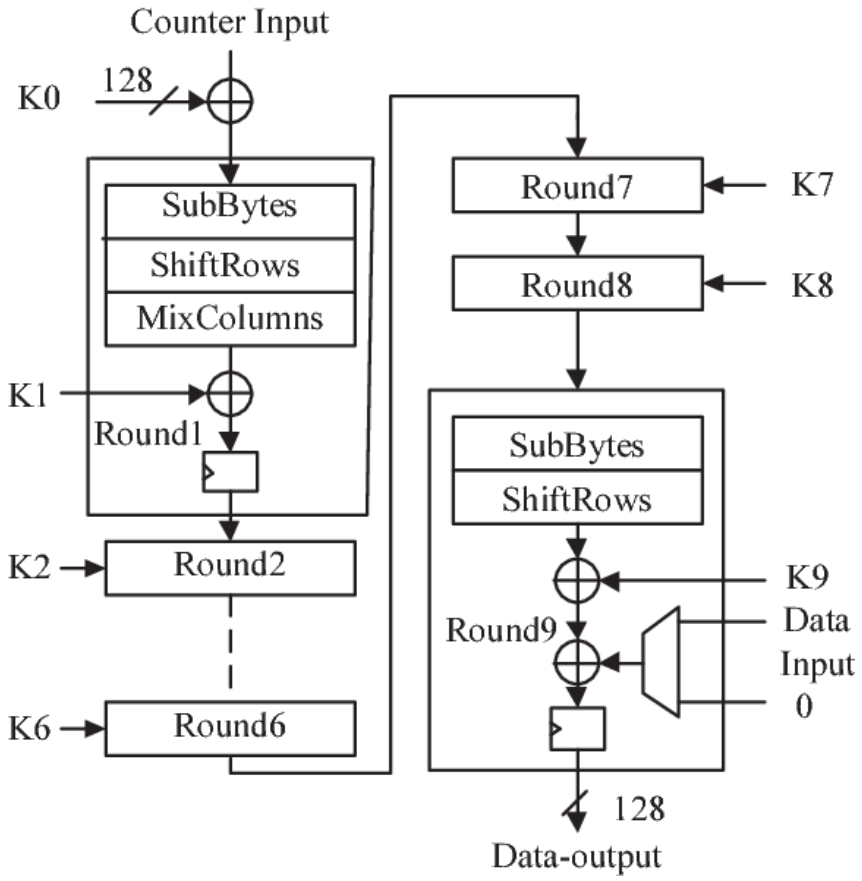

Figure 3.4. The pipelined AES architecture [37].

3.1.2. AES Algorithm Working Steps

General steps of an algorithm for 128-bit block encoding and decoding [30].

1. By the length of the encryption key, the number of rounds used to repeat the encryption techniques, are extracted within the algorithm.
2. A state array is prepared depending on the data entered into the block.
3. The first-round key is prepared and sent to the start state array.
4. Executing the number of supposed rounds, except the last round to manipulate the data entered into the state.
5. Implementation of the last round to end the process of manipulating the state's data.
6. The data inside the state is copied and extracted as ciphertext.

Steps of AES algorithms: Each round of the encryption process requires a series of steps to change the state of the array [34].

Special techniques that are used for encryption within the algorithm [38, 39]:

1. Block to state: The picture's data is converted inside a block to an array 4*4 called state, after converting the text values to hexadecimal.

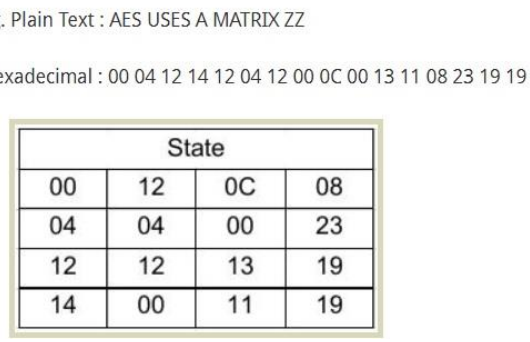

Figure 3.5. Block to state operation.

1. Add round key transformation: Doing XOR process between the state and the value of key after converting it into an array.

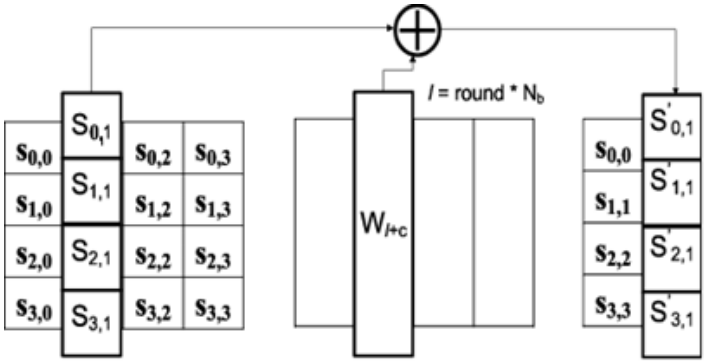

Figure 3.6. Add round key operation [24].

3. Sub Bytes transformation: Replace bytes transformation (what matched in the table) bringing out a new state after matching the resulting state with the S box table.

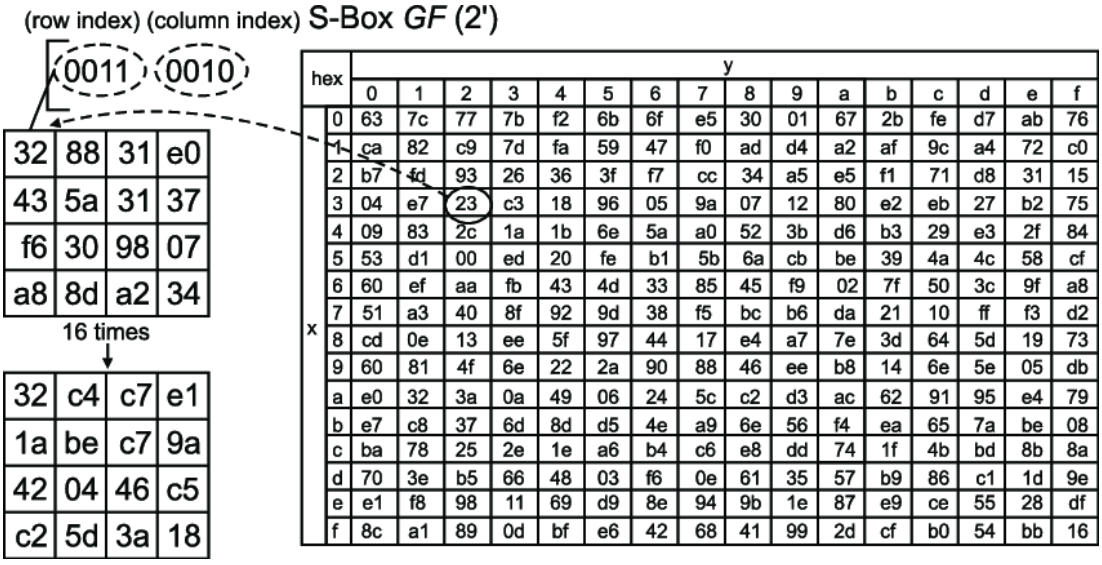

Figure 3. 7.Sub bytes transformation [40].

4.Shift rows transformation: Each row is rotated to the right by a certain number of bytes according to the Figure 3.7, rotation starts in the second row.

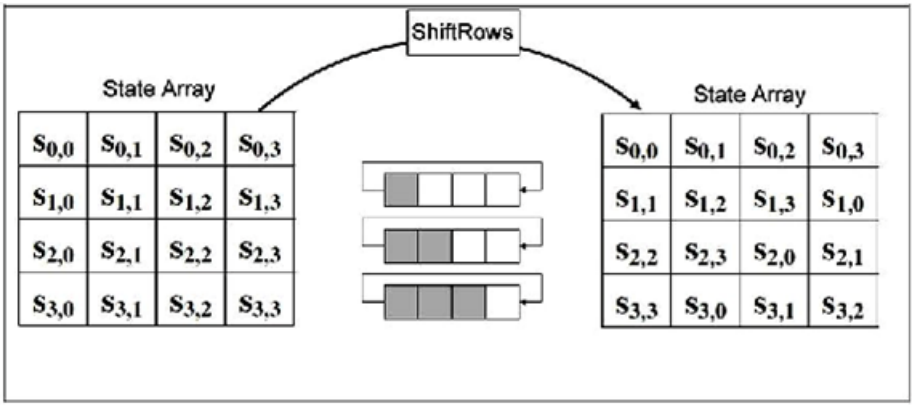

Figure 3.8. Shift rows transformation [24].

5. Mix columns transformation: The resulting state is multiplied by a fixed-value mix columns array; the resulting value is the ciphertext.

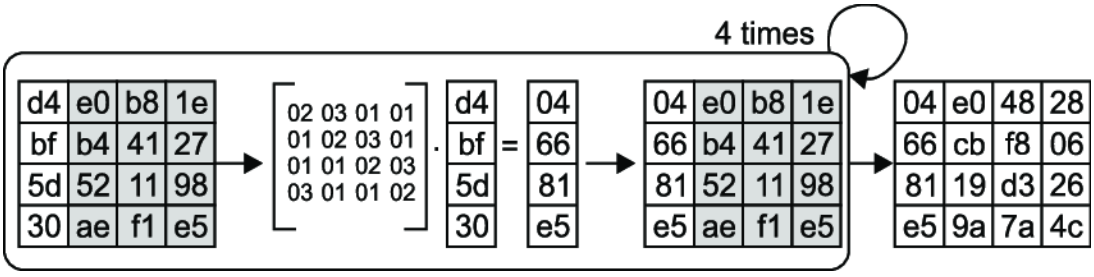

Figure 3.9. Mix columns transformation [40].

Decryption**:** Inverses all steps are taken in the encryption process, using the inverse functions [34], as found in Fig 3.9.

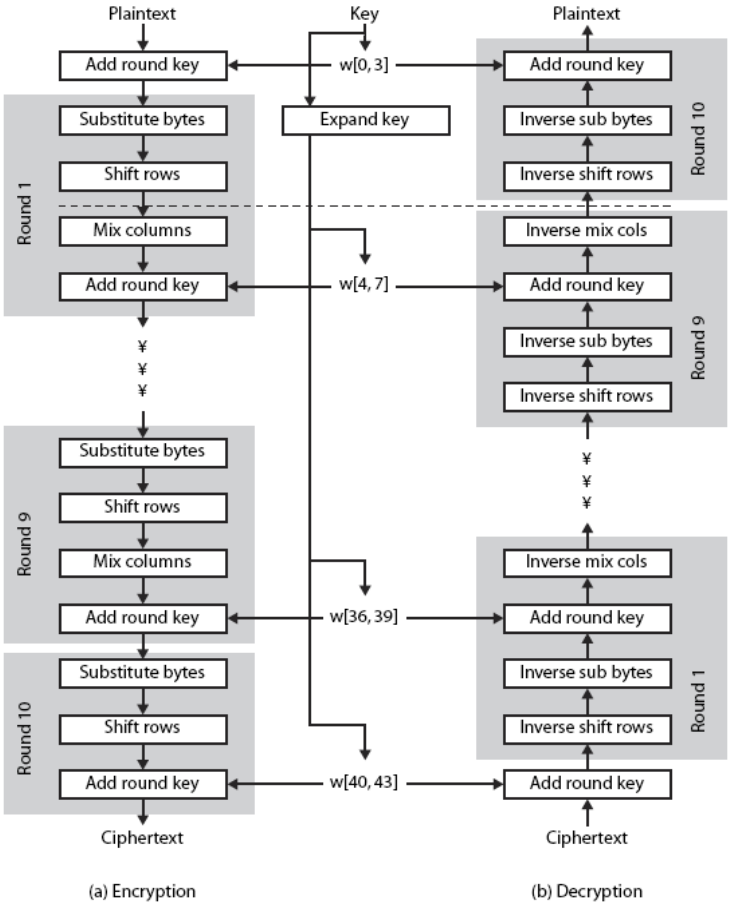

Figure 3.10. AES encryption and decryption operations [41].

The pseudocode of the AES encryption process is given below [42].

INPUT **STATE**, **KEY**
OUTPUT **CIPHER**
State=AddRoundKey (STATE, KEY [0…,3]);
FOR i =1 to Round
{
STATE = Substitute Bytes (STATE);
STATE = Shift Rows (STATE)
IF (i < Round)
STATE = Mix Columns (STATE)
STATE =AddRoundKEY (STATE, KEY [4*i, …., 4*i+3]);
}
CIPHER=STATE

3.1.3. Features of AES Algorithm

Advantages of AES algorithm [43]:

1- The algorithm has three key lengths that prevent the hacker from guessing its values, as it may take millions of years to complete the experience of all possibilities for the weakest key value, which is 128 bits.
2- The attacker loses a great cost to the hacking process because of the key and takes a time of about hours or days.
3- There is no weak point to attack.
4- The algorithm has great and high computational power in theoretical research.
5- Design details and specifications are complete.
6- Symmetric key symmetric block cipher.
7- 128-bit data, 128/192/256-bit keys.
8- Stronger and faster than RSA, DES, Triple-DES.

3.1.4. Difficulties and Constraints of AES

Difficulties of AES [44, 45]:

1- The AES algorithm suffers from CPU size and thus leads to low resource consumption.
2- The efficiency and speed of the algorithm in implementation are not at the required level.
3- Weak flexibility, as analysts mentioned that having one copy of the algorithm facing attacks and protecting products may expose data to danger. Another copy must be added to protect products in case the first copy is broken [46, 47].
4- Weak security is widespread. Researchers have stated that high security and high efficiency must be provided to an unprecedented extent.
5- Intellectual property and implementation costs. Some researchers have argued that if there is more than one version of the algorithm. It will increase the costs of production and implementation. Also, the other obstacle is the intellectual property rights for this algorithm.

3.1.5. Solutions to AES Obstacles

In the problem of algorithm speed and improvement, researchers have added a parallel computing technology with the algorithm to create a fast-encoding system based on GPU parallel computing. The approach has proven its effectiveness as they were able to greatly accelerate the work of the encryption algorithm [45]. Non-multiple copy with high efficiency to reduce production costs, while monitoring its work in securing products [48].

The solution to the issue of the intellectual property of the algorithm, is to choose one algorithm with improvement. It will enhance interoperability and address vendor concerns about implementation costs and intellectual property if the number of the algorithm is increased[44].

3.2. TRIPLE DES (3DES)

Known as Triple Data encryption Algorithm (TDEA). It's a type of symmetric block cipher, developed in 1998 to replace the DES algorithm, due to its apparent errors and the shortness of the key used [49, 50]. DES algorithm key value is no longer appropriate in the face of modern techniques in cryptographic analysis and supercomputing. Therefore, the value of the key must be increased as in 3DES to make the ciphertext safer. It contains 64-bit block size with a 168-bit key size three times the value of DES (3*56) [32]. Concerning the working steps of the algorithm, it is the same as the DES method [51], but the encryption process is repeated with the mentioned steps three times depending on the three keys of the 3DES algorithm.

3.2.1. 3DES Architecture

3DES is the same as for the DES algorithm. The encryption process goes through three steps. The first key is used to encrypt the plaintext, and the resulting ciphertext decrypt by k2, and k3 will encrypt the resulting text after k2 decryption. The output is the ciphertext. The decryption operation is the opposite of the encryption operation. Decryption is done through third key and the resulting text is encrypted with the second key, and the ciphertext is decrypted to become the plaintext [32, 50], as shown in the

Figure 3.10. The three steps listed are taken to increase the complexity of the ciphertext, and will take more time in the encryption process.

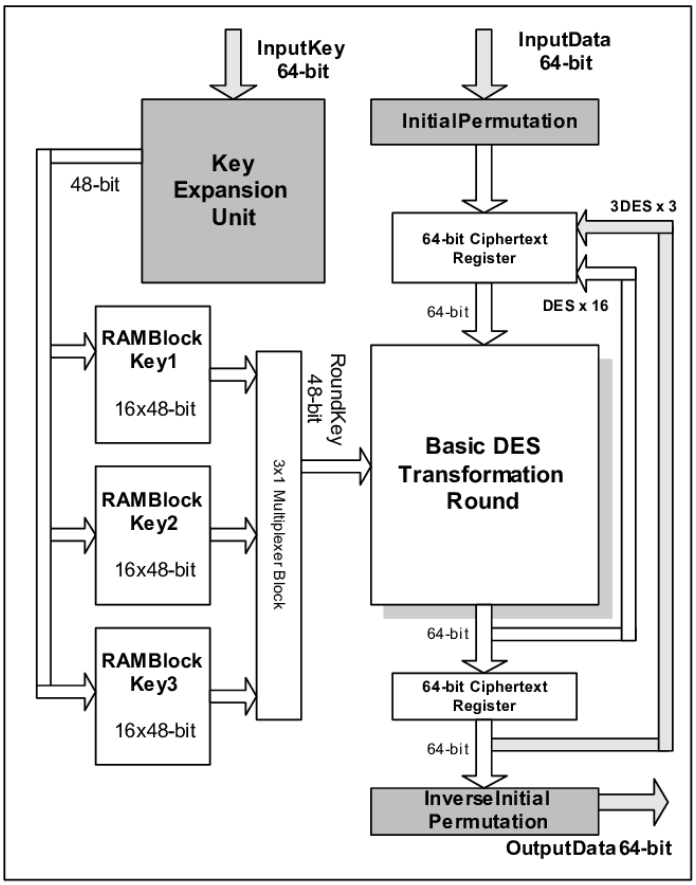

Figure 3.11. 3DES encryption and decryption process [52].

3.2.2. Advantages of 3DES Algorithm

Advantages of 3DES algorithm [53]:

1- The algorithm has good security and ease for implement operations.
2- One DES encryption can be executed if the three key values are equal to meet the needs of some applications that support a single DES.
3- Very effective in devices and all hardware.
4- It is used well in financial systems.
5- Effective on protecting the biometric information in electronic passports.
6- It is very easy to modify software programs to use 3DES.

7- Key length eliminates more attacks that try to reduce the time taken to break 3DES algorithm.
8- It has the advantage of performing reliability.

3.2.3. Disadvantages of 3DES Algorithm

Disadvantages of 3DES algorithm [50, 53]:

1- It is not completely secure and cannot protect data for a long time.
2- A vulnerability in addressing security due to the presence of three independent keys when configuring it may encounter a meet-in-the-middle attack.
3- Security flaws that allow the hacker to obtain key length, thereby reducing the time to crack the key.
4- It is not particularly successful when it comes to encrypting big messages.
5- Unsecure transfer of the key between client and server.
6- Slow to use and encryption due to having three keys and too much duplicate encryption operations.

3.3. COMPARISON BETWEEN AES AND 3DES

Researchers concluded in [53], that AES is better in terms of security and that it is unbreakable in practical use. 3DES has been exposed to attacks and breaches. AES algorithm increases the strength of computers against piracy attacks by having great encryption complexity, it prevents unauthorized persons from obtaining data. Also, AES provides a source of security to mobile devices for long-term. AES algorithm has made progress over 3DES in the field of data protection.

3.3.1. Computational Time Analysis

We executed the AES and 3DES algorithm ten times for encryption and decryption on the three satellite images with different sizes. Also, we calculated the elapsed time for each of them. The average calculation times are found of the two algorithms for both encryption and decryption. As in Table 3.1, the computational times of AES algorithm for all images are less than 3DES in both the encoding and decoding process. We have

concluded that, the encoding process in the AES algorithm reaches approximately 40% of the 3DES, and this proves that the AES algorithm is superior to the 3DES at the implementation time.

Table 3.1. Computational time.

Image	Encryption Time of AES	Encryption Time of 3DES	Decryption Time of AES	Decryption Time of 3DES
Small	0.00606	0.01197	0.01385	0.01930
Medium	0.00955	0.02480	0.02327	0.04417
Large	0.02016	0.08411	0.06405	0.14450

PART 4

APACHE SPARK

Apache Spark is a cluster computing platform and open-source project that was established in 2009. It is used for big data analytics and is designed to work quickly with general purposes. Also, it works to expand the map-reduce environment. The factors that give Spark speed and efficiency, is the ability to perform computations in memory [54].

The components of Spark:

1- Spark Core includes a set of basic functions such as memory management and interaction with storage systems [55].
2- Spark SQL is one of Spark packages that contains the processes for handling and organizing data set [56].
3- Spark Streaming, it is a component of Spark that processes live data streams as well as log files created by a web server [57].
4- Machine Learning Library (MLib) is a library inside Spark that contains a group and multiple types of Machine Learning algorithms and contains machine learning functions [58].
5- GraphX is one of the libraries within Spark that specializes in creating random graphs and processing these graphs [59].
6- Cluster Managers is used to increase the efficiency of the Spark environment by increasing the number of mathematical nodes from block manager [54]

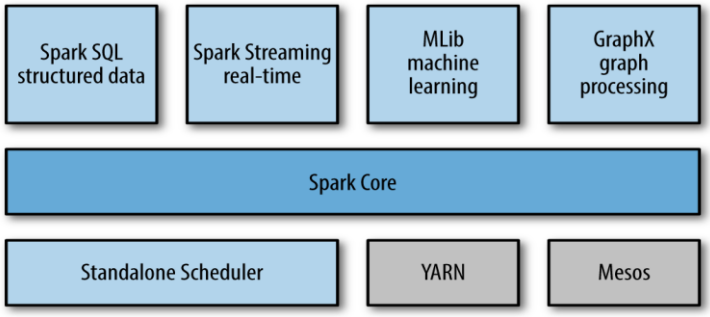

Figure 4.1. Apache spark components [54].

Spark's framework revolves around the general purpose of cluster computing. It is used in many applications that are classified into two categories for typical use cases, which is the category of data science and data applications [54].

4.1. RESILIENT DISTRIBUTED DATASET (RDD)

RDD is a set of flexible data elements distributed on a group of devices. It allows users to store data on the hard disk or in memory. Also, RDD allows data sharing with high capacity across computations [60]. It is the main and important factor in Spark, which will be used and relied upon the encryption algorithm to distribute data.

The process of storing and sharing data with MapReduce is slow due to the repeated copying and adding data to the hard disk. Spark solved slow issue by storing the results in distributed memory instead of storing them on the hard disk. So, the system becomes faster, and if RAM is insufficient, the rest of the distributed data is stored on the hard disk [56, 60].

4.2. COMPARISON SPARK AND HADOOP

Spark and Hadoop evaluated by analyzing files in the context of cloud computing. The results prove that Spark is faster and more flexible due to the presence of RDD which distributes data to the memory instead of the hard disk. Spark outperformed Hadoop in the execution time factor [61]. As Spark reduces the execution time, it was

discovered that Spark uses higher resources than Hadoop in processing operations. Spark by saving on memory was higher than Hadoop [62], as shown in the Figure 4.2.

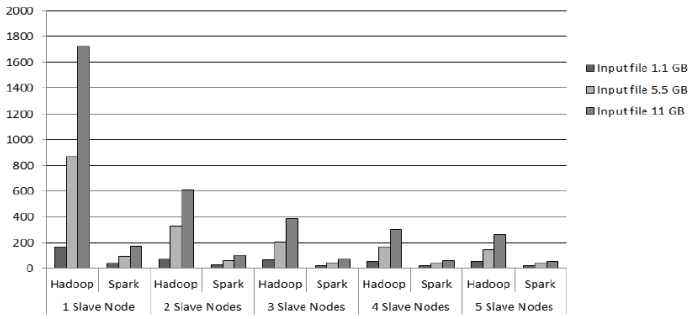

Figure 4.2. Execution time of hadoop and spark [62].

4.3. AES ALGORITHM WITH HADOOP IN CLOUD

The data is encrypted when it is stored online via cloud technology. Apache Hadoop is used to distribute and store data temporarily during processing. Three DES, RSA, and AES algorithms were selected for encryption, and make a comparison to measure the ability of these algorithms within cluster. The results appeared in Figure 4.3, that the AES algorithm is more efficient than the rest in terms of measuring execution time, and the volume of data that consumes less space in storage [20].

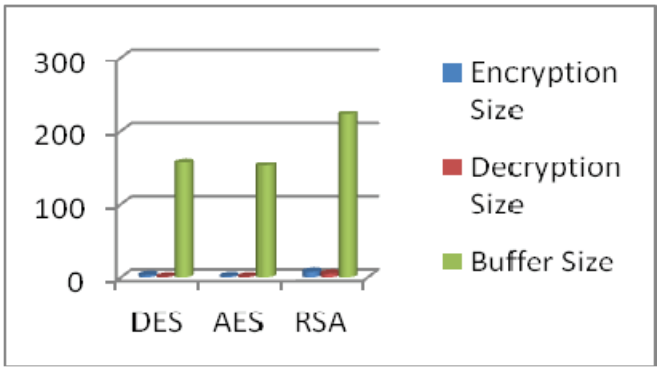

Figure 4.4. Comparative status of encryption algorithms for packet size 153[20].

Packet Size(KB)	153		
	DES	AES	RSA
Encryption Time(Sec)	3.0	1.6	7.3
Decryption Time(Sec)	1	1.1	4.9
Buffer Size	157	152	222

Figure 4.5. Comparison of DES, AES & RSA algorithm for packet size 153 in execution time [20].

Packet Size(KB)	868		
	DES	AES	RSA
Encryption Time(Sec)	4.0	2.0	8.2
Decryption Time(Sec)	1.8	1.2	5.1
Buffer Size	888	889	934

Figure 4.6. Comparison of DES, AES & RSA algorithm for Packet size 868 in execution time [20].

4.4. AES ALGORITHM WITH SPARK IN CLOUD

In [7] Li studied the encryption of video because it contains huge data, by cutting it into small images, and in coordination with the cloud platform. Li used the AES encryption algorithm with iframe and MV chips, to give the encryption information and control the depth of the encryption. The encryption technology has been adapted to the Apache Spark environment inside the cloud. Spark has the ability to distribute and store data. The encryption was done in two ways, the first is selective regular encryption with one server. The second way is selective encryption with Spark inside the cloud. Through the results it was found that selective encryption for one server has relatively low security, weak efficiency, and slow execution time. About selective encryption with Apache Spark, he found an improvement in the efficiency of the encryption relatively while meeting the requirements of time and high security as in Figure 4.6, Spark contains one master and two nodes. Thus, he concluded that Spark could significantly improve the ability and capability of the encryption process, meet

real-time requirements, and high performance. Also, it is better to use Spark with encryption inside the cloud than any other environment.

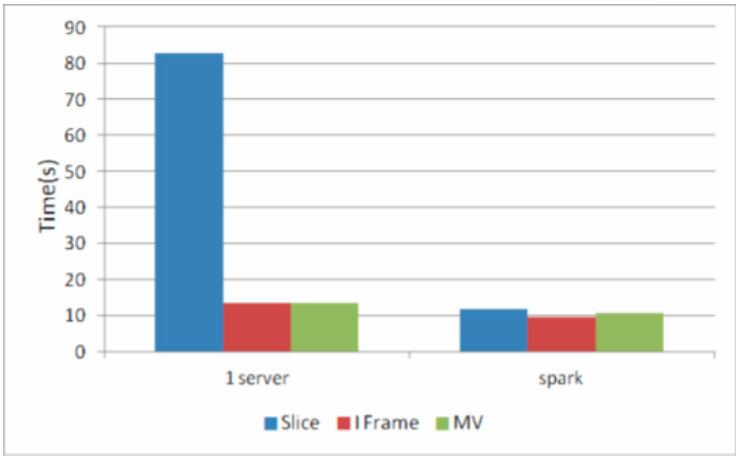

Figure 4.7. Comparison between 1-server and spark [7].

PART 5

SATELLITE IMAGE ENCRYPTION WITH SPARK

In this chapter, we mentioned the practical side and general steps to encrypt and decrypt the satellite images using the AES algorithm with Apache Spark environment. We are giving examples for the results of each process, mentioning the methods used and techniques to adapt the algorithm inside the cloud. Python programming language used in this thesis on the Jupiter Notebook 6.0.1 platform with the Apache Spark environment. The thesis work diagram is shown in Figure 5.1.

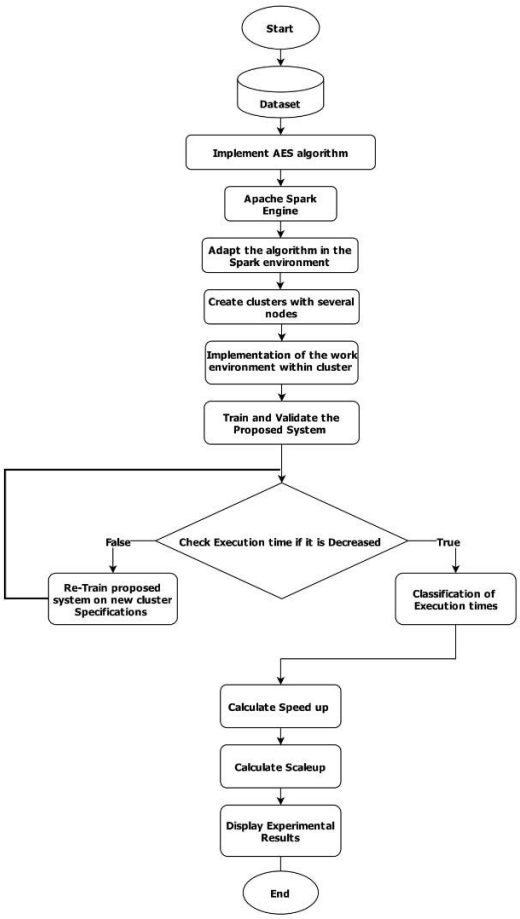

Figure 5.1. The thesis work diagram.

5.1. APACHE SPARK ARCHITECTURE

On the practical side of Apache Spark, it works by operators as independent sets of processes within a cluster. The operations are coordinated by an object called Spark Context within the main program (driver program). Spark Context contacts various types of cluster managers (either Spark's own Standalone cluster manager, Mesos or YARN). Job of Spark Context is to allocate resources across applications. Spark gets the executors of the nodes in the cluster. The executors represent the processes for running computations and storing data. The next step is to send the application or task code (defined by JAR or Python files passed to Spark Context) to the executors. Finally, Spark context sends tasks to the executors for implementation and processing [54].

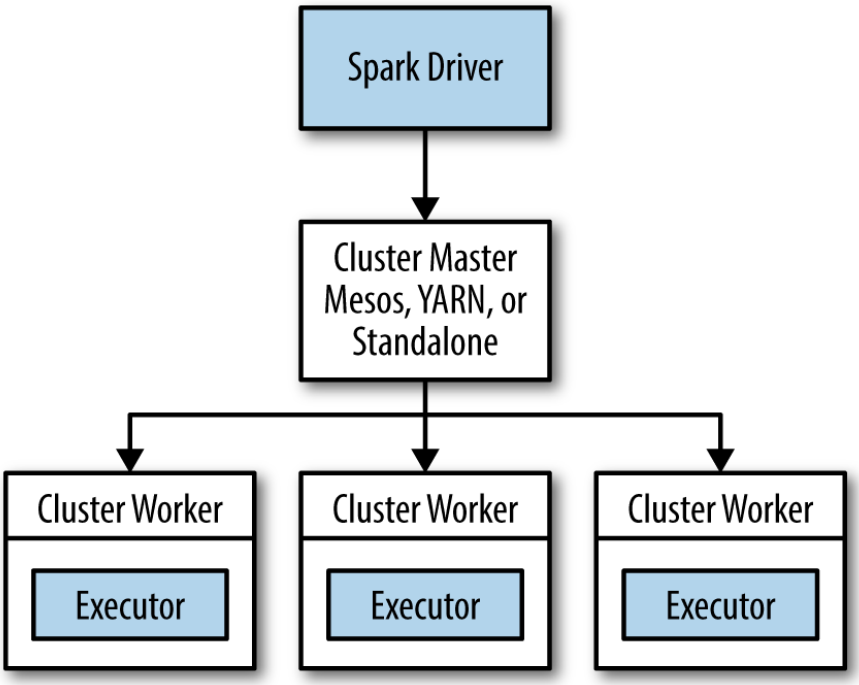

Figure 5.2. The components of a distributed spark application [54].

5.2. IMPLEMENTATION STEPS OF STUDY

The basic steps of this study are divided into groups as shown in Figure 5.3.

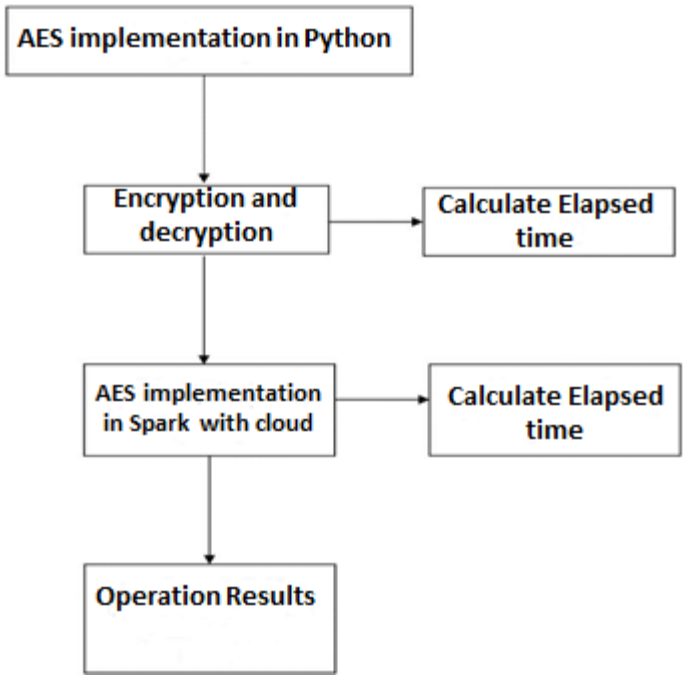

Figure 5.3. Process steps.

5.2.1. First Step

In the first step, we implemented the AES algorithm in the Python language. We encrypted the satellite images through some commands that convert the image pixels into bytes. Then, the satellite images are decrypted by reversing the encryption process. The execution time for each of the two processes is calculated as shown in the Figure 5.4, it displays the times of encryption and decryption of satellite image.

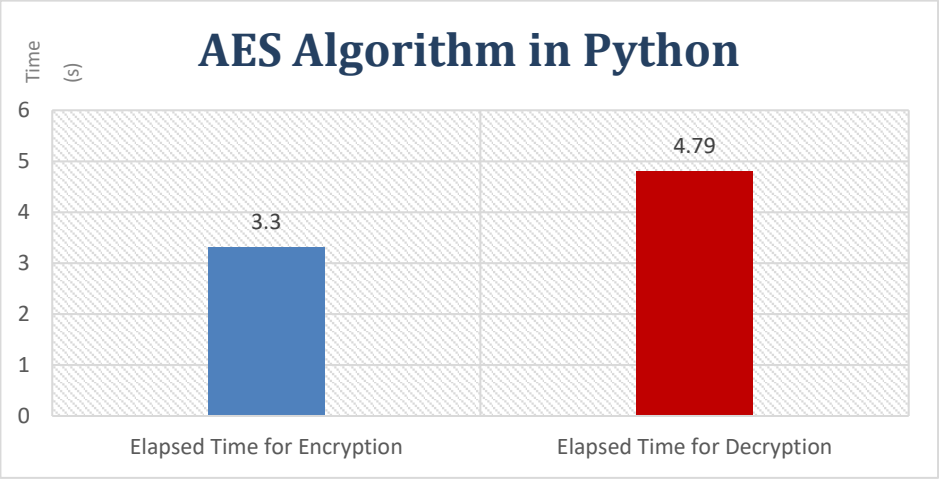

Figure 5.4. Execution time for encoding and decoding one satellite image.

5.2.2. Second Step

In the second step, AES algorithm is adapted to Apache Spark to perform the encryption and decryption process in standalone mode inside the cluster. This was done in the Jupiter platform by installing the Pyspark library. The command also needs to install spark-3.0.1-bin-hadoop2, and the jre1.8.0_271 Java program for the latest version inside the personal computer. After successfully adapting the AES algorithm inside Apache Spark, we encrypted satellite images and measured the elapsed time as shown in the Figure 5.5.

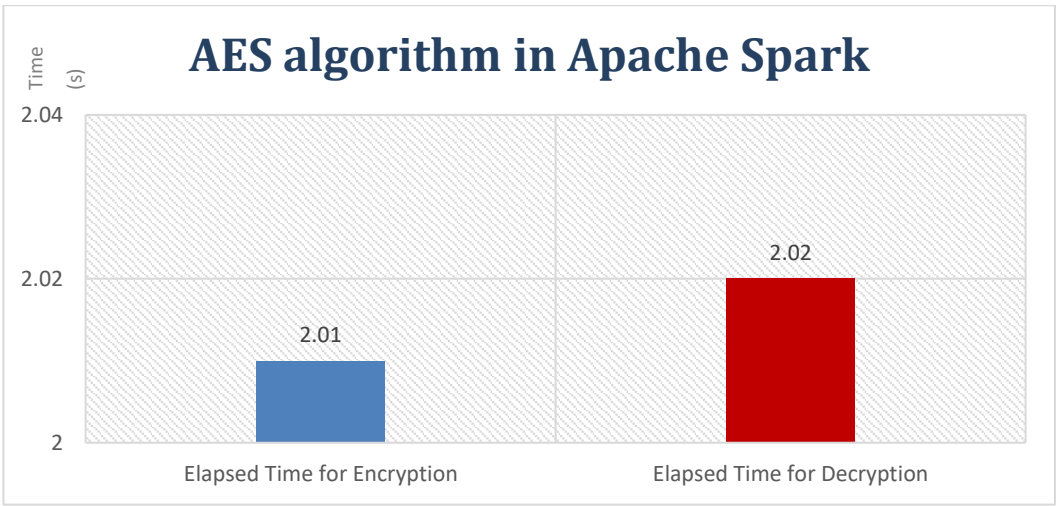

Figure 5.5. Execution time for AES encyption and decryption of satellite image in Apache Spark standalone mode.

5.2.3. Third Step

The algorithm is implemented with Spark inside the cloud. The Google cloud platform service is used because it supports jupyter notebook. A cluster is created with jupyter notebook. We upload the program file to encrypt satellite images with Apache Spark from the personal computer and, upload the satellite images to the Google cloud store. Figure 5.6 shows the encryption and decryption time of satellite image inside the cluster.

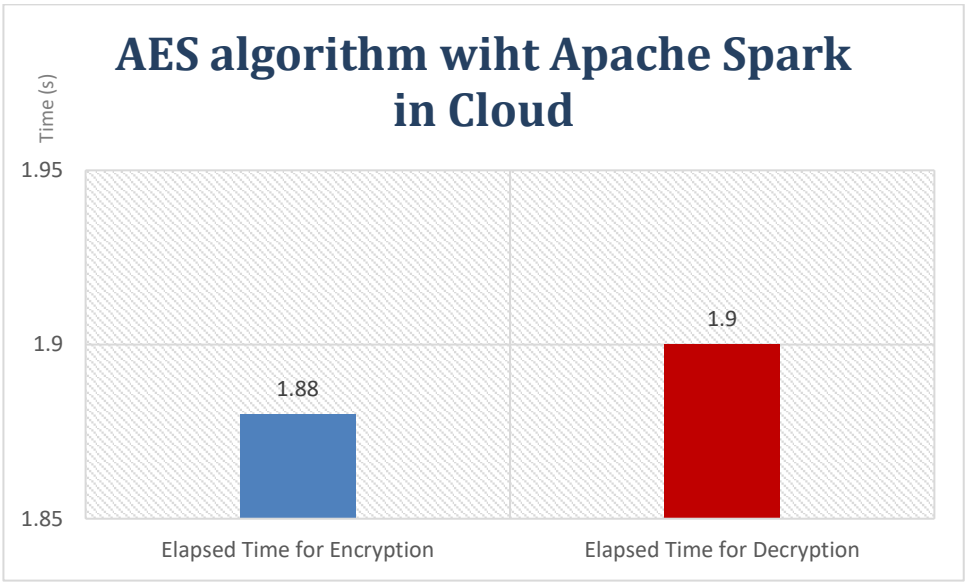

Figure 5.6. Execution time for AES encryption and decryption of satellite image by Apache Spark in cloud.

5.2.4. Fourth Step

In this step, the cluster specifications are increased during the configuration. We increase the number of nodes and the CPU cores. Encryption and decryption process continue on the image until we get the least time possible based on the high specifications. Thus, we gain the maximum speed factor in real-time execution (Figure 5.7).

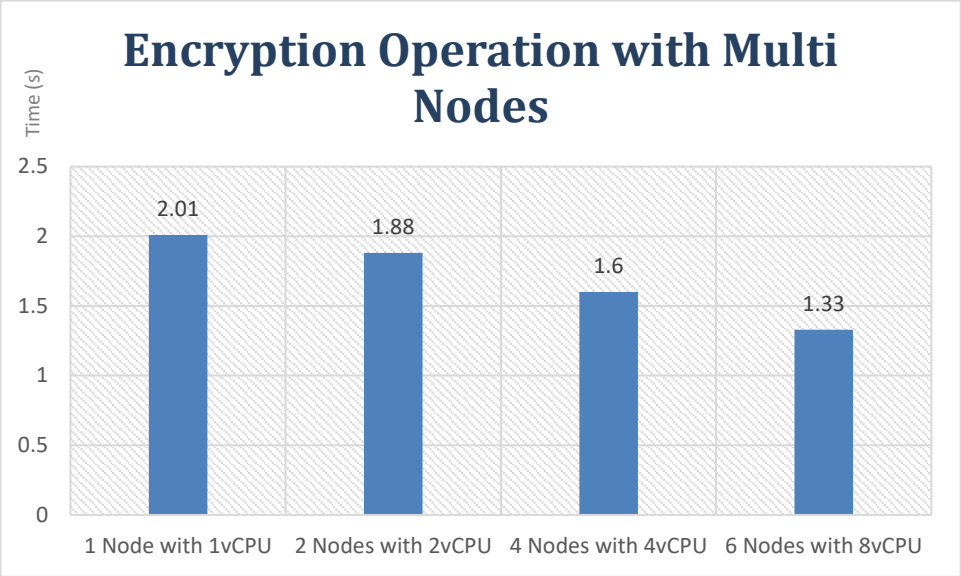

Figure 5.7. Encryption operation.

5.2.5. Fifth Step

In the last step, after the success of the encryption technology on both the Apache Spark environment and the cloud platform, huge data chunks of satellite images are encrypted and decrypted. To measure the ability of AES algorithm, we calculated the percentage of Speed up (Sp) and Scale up (Sc). Speed up can be defined as the ratio between execution time for one processor with one node, over the ratio of runtime for multiple processors and nodes. It is affected by the number of nodes and specifications of the CPU whenever the number of nodes increasing. While the value of speed up is decreased, the improvement in performance gained using multiple processors is measured. It is calculated with the formula as shown bellow [63].

$$Sp = Ts/Tp \tag{5.1}$$

Where Ts represents the execution time for a single node, and Tp represents the execution time for multi nodes(p-nodes).

Scaleup is used to measure and deal with the excessive volume, by increasing the capacity of the disk and processors, and the number of nodes to deal with the additional load. Its work is to maintain the response time or implementation is fixed with

increasing the volume of data, by adding processors, disks and additional nodes, the mathematical formula for it shown bellow [64, 65].

$$Sc = Ts/Tn \tag{5.2}$$

Where Ts represents the execution time for a single node and Tn its execution time represents the increase in the data set size for the satellite images. Depending on the results of the previous steps, we see that the execution time for all encryption operation is less than the execution time for the decryption operations. Also, the results of the execution time in the standalone mode are less than the AES by python, because cluster for stand alone mode provides one node that act as both master and worker which is slightly faster than AES algorithm on python. AES in the standalone mode is larger than the results of the AES algorithm in the cloud with multi nodes. We conclude that the speed factor was achieved by cloud, and the execution time of the AES algorithm was reduced twice to the minimum execution time for the encryption and decryption processes on satellite images. We achieved the time gain feature with real-time execution, as shown in the Figure 5.8.

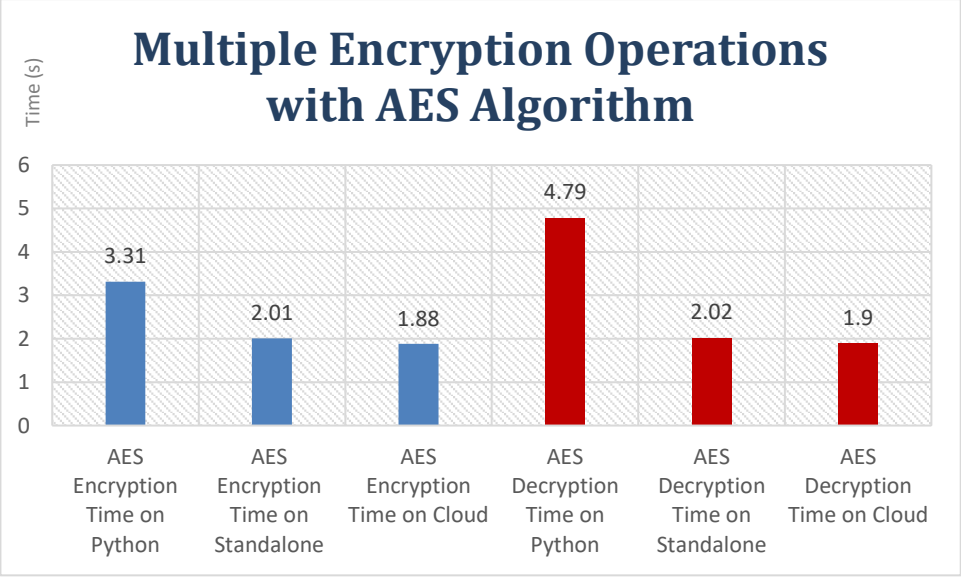

Figure 5.8. Several encryption operations with AES algorithm.

PART 6

EXPERIMENTAL RESULT

In order to discuss the work of the thesis and show the conclusions, we presented the results of the practical side in tables and graphs. We mentioned the data set used in detail as well.

6.1. DATASET DESCRIPTION

For this dissertation, the satellite images data used in this thesis captured by the SAS Planets version (15.11.11.9233). The dataset size is 7.37 GB (7,916,688,842 bytes), it can be found on (https://www.kaggle.com/mohamma/karabuk-city-satellite-images). It is almost 5,551 images with dimensions 1909*1662pixels and size 1.05 MB (1,106,025 bytes) for most of the pictures. Because storage capacity is limited within the Google Cloud Platform, 4 GB (approximately 3 thousand images) of the full size was used in the encryption and decryption process inside the cluster. The dataset representing pictures of various locations in the city of Karabük, inside Turkey. In the basic degree, we upload the data to the Cloud Platform Store. Also, we will take the path of the images. The data of each image will be converted into bytes to be entered in the encryption stage with Apache Spark. We used satellite image data of a large size, to measure the elapsed time and gain more time. When increasing the specifications of the cluster and nodes, we reached the least time possible for the encryption and decryption process.

6.2. EXECUTION TIME RESULTS

Table 6 shows, the encryption and decryption processes of satellite images with a size of 4 GB, from the single nodes to 16 nodes in the cloud with CPU specifications

Table 6.1. Execution time results for AES algorithm with Apache Spark in cloud.

Node Numbers	CPU Specifications	Encryption Time (sec)	Decrption Time (sec)
Single Node	1 vCPU, 3.75 GB memory	880.33	880.41
2	2 vCPU, 7.5 GB	472.57	472.49
4	4 vCPU, 15 GB	300.59	300.57
6	8 vCPU, 30 GB	170.55	170.52
8	16 vCPU, 60 GB	120.95	120.52
10	32 vCPU, 120 GB	90,80	90.10
12	64 vCPU, 240 GB	72.79	72.41
14	96 vCPU, 360 GB	62.69	62.25
16	n1-highmem-8.8 vCPU, 52 GB	54.88	54.12

Table 6.1 and Figure 6.1 shows the decrease in execution time as the number of nodes is increased.

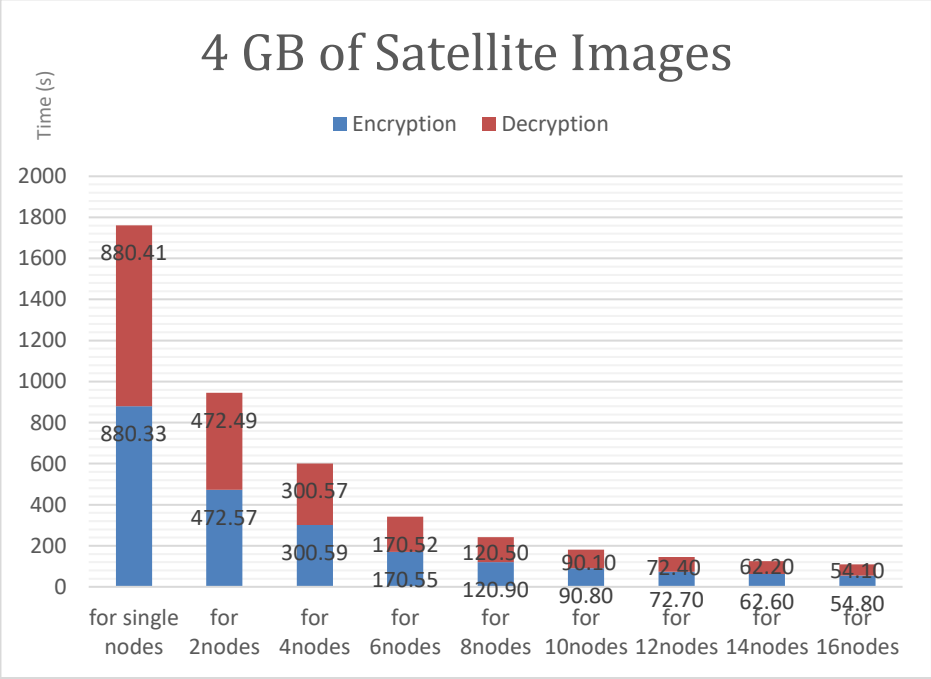

Figure 6.1. Implementation times for the AES algorithm work.

6.3. IMPLEMENTATION RESULTS FOR SPEEDUP

Speedup can be calculated by **Sp = Ts/Tp,** where Ts represents the execution time for a single node, and Tp represents the execution time for multi nodes.

Table 6.2. Speed up value for encryption and decryption.

Ts/Tp	Speed up for Encryption	Speed up for Decryption
single node/2 node	1.86	1.86
single node/4 node	2.92	2.92
single node/6 node	5.16	5.16
single node/8 node	7.27	7.30
single node/10 node	9.69	9.77
single node/12 node	12.09	12.15
single node/14 node	14.04	14.14
single node/16 node	16.03	16.26

Depending on the figure below, linear results, and high time gain are shown

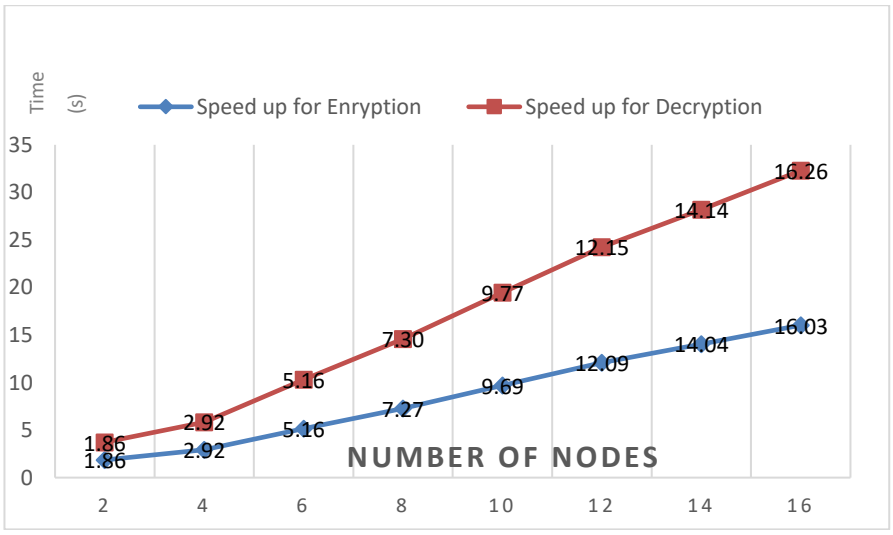

Figure 6.2. Value of speed up.

6.4. IMPLEMENTATION RESULTS FOR SCALEUP

Scaleup can be calculated by **Sc = Ts/Tn,** where Ts represents the execution time for a single node and Tn its execution time represents the increase in the data set size for the satellite images and the increase in the node's value.

Table 6.3. Execution time.

Number of Nodes	Data Size	Encryption (sec)	Decryption(sec)
1	250MB	40.67	40.68
2	500MB	75.55	75.56
4	1 GB	124.19	124.20
8	2 GB	120.95	120.52
16	4 GB	54.88	54.12

Table 6.4. Scale up value for encryption and decryption.

Ts/Tn	Encryption	Decryption
1Nodes with (250MB)/2Nodes(500MB)	0.53	0.53
1Nodes with (250 MB)/4Nodes (1 GB)	0.32	0.32
1Nodes with (250 MB b)/8Nodes (2 GB)	0.33	0.33
1Nodes with (250 MB)/16Nodes(4GB)	0.74	0.75

The converging result of scale up ranges from 0.3 to 0.7, as we can see in the Figure 6.3. It was found that the encoding value is less than the decoding value.

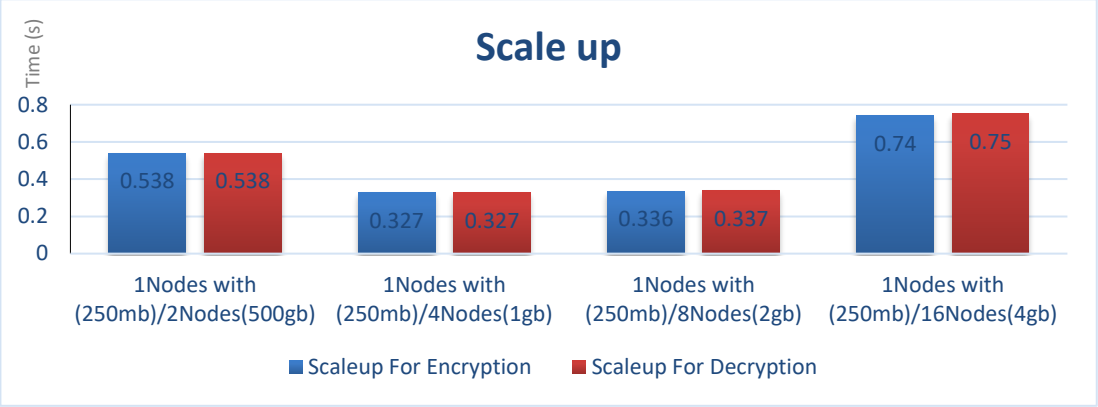

Figure 6.3. Value of scale up.

6.4. CHALLENGES

Some of the challenges in the work were mentioned:

- When AES algorithm is built in Python language, there was difficulty in the image. The images are encrypted as a file, not as a picture. The problem here, images cannot be displayed to see its details. After encrypting images as a file, the file cannot be opened. We found a method (Image.frombytes) to read the image in bytes; Thus, the encrypted image is displayed and decrypted.

- To implement the algorithm inside Apache Spark, it was necessary to use the RDD to read the image data and distribute it to the nodes to gain more time from encoding. At first, the encryption and decryption process on the satellite images failed with RDD. We concluded that the RDD reads the image path and sends it with the rest of the image data to the algorithm for processing. The encryption and decryption failed. So, we separated data from its path by a special matrix and sent to the algorithm to perform the encoding and decoding process.

6.5. DISCUSSION

The speed of the algorithm's performance in stand alone mode was 880 seconds as shown in Table1 after increasing the number of nodes, the execution time seemed to be reduced by almost half at 16 nodes, where the time became 54.886 to encode the data. So, the increase in time gain is approximately 65% of the actual time for the encryption and decrryption process on satellite images. The Speed up experiment results, as shown in Figure 6.2, follow the ideal acceleration with AES algorithm. The acceleration value is affected by increasing the number of nodes and CPU. As shown in Figure 6.3, Scale up result show that the algorithm's expansion capability is fairly close. The expansion system is balanced and is not affected by the increased volume of data used for encryption and decryption.

PART 7

CONCLUSION

In this research, we encrypted huge data chunks of satellite images via the AES algorithm in Apache Spark. The encryption and decryption process were implemented on the data of satellite images inside cloud. In order to evaluate the ability of AES algorithm, we used the Speed up and Scale up ratios. After comparing the elapsed time result of the AES algorithm with the stand-alone mode, and inside the cloud, we concluded that the algorithm became faster inside the cloud and a positive runtime was gained. In Scale up, to assess the scalability of the algorithm, we used a set of satellite image data. After changing the volume of data and performing the encryption process, we found that the results are close, and the expansion system is stable and not affected by the process of increasing the data volume. In Speed up, the acceleration value was perfect and very close to linear acceleration. So, we conclude that the ability and durability of the algorithm, and its efficiency in encoding satellite images have become much higher and, the system is balanced.

We got conference paper as a result of thesis:

- Yasin Ortakci and Mohammed Yaseen Abdullah Al-Hayani, Performance Analyses of AES and 3DES Algorithm for Encryption of Satellite Images, 5th International Conference on Smart City Applications

In future plans, we aim to implement the algorithm on other real-world applications. Also experiment with encryption using the algorithm on huge data, with the use of high specifications of cloud computing and a large number of nodes. We need to conduct comprehensive experiments on encryption operations for big data and see the results.

REFERENCES

1. Bhargavi, I., D. Veeraiah, and T.M. Padmaja, "Securing BIG DATA: A Comparative Study Across RSA, AES, DES, EC and ECDH", ***Lecture Notes in Networks and Systems*** 5 (5): 355-362.(2017).

2. Tan, C.K., et al., "Security protection of DICOM medical images using dual-layer reversible watermarking with tamper detection capability", ***Journal of Digital Imaging***, 24 (3): 528-540.(2011).

3. Upadhyay, P. and S.J.I.-. Gupta, "Introduction to satellite imaging technology and creating images using raw data obtained from landsat satellite", ***International Conference on Glass Technology and Innovations***, 1 (1): 126-134.(2012).

4. Bodhale, A. and J. Kulkarni, "Satellite Imagery Tools"; **https://www.researchgate.net/profile/Jyoti_Kulkarni3/publication/315698706_Satellite_Imagery_Tools/links/5bd01208a6fdcc204a03638b/Satellite-Imagery-Tools.pdf** (2017).

5. Kumari, M., S. Gupta, and P.J.D.R. Sardana, "A survey of image encryption algorithms", ***3D Research***, 8 (4): 37.(2017).

6. Karthigaikumar, P., S.J.I.s.i.o.c.s.-n.d. Rasheed, and p. NCCSE, "Simulation of image encryption using AES algorithm", ***IJCA special issue on computational science-new dimensions & perspectives NCCSE***, 8 (4): 166-172.(2011).

7. Li, C., et al. "A video selective encryption strategy based on spark", ***IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)***, IEEE, , China,757-760 (2016).

8. Shrivastava, A. and A. Tiwary "A Big Data Deduplication Using HECC Based Encryption with Modified Hash Value in Cloud", ***Second International Conference on Intelligent Computing and Control Systems (ICICCS)***, IEEE, Madurai, India,484-489 (2018).

9. Aljarah, I. and S.A. Ludwig "Parallel particle swarm optimization clustering algorithm based on mapreduce methodology", ***Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)***, IEEE, Mexico City, Mexico,104,111 (2012).

10. Canbay, Y. and S. Sağıroğlu "Big data anonymization with spark", ***International Conference on Computer Science and Engineering (UBMK)***, IEEE, Antalya, Turkey,833-838 (2017).

11. Shah, S.Y., B. Paulovicks, and P. Zerfos "Data-at-rest security for Spark", ***IEEE International Conference on Big Data (Big Data)***, IEEE, Washington, DC,1464-1473 (2016).

12. Al Mamun, A., et al. "BigCrypt for big data encryption", ***Fourth International Conference on Software Defined Systems (SDS)***, IEEE, Valencia, Spain,93-99 (2017).

13. Chen, S., W. Hu, and Z. Li "High Performance Data Encryption with AES Implementation on FPGA", ***IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)***, IEEE, Washington, DC, USA,149-153 (2019).

14. Yang, Z., et al. "A Distributed Video Encryption Method Based on Spark", ***DEStech Transactions on Computer Science and Engineering***, DEStech Publications, China,4 (2017).

15. Mehmood, M.S., et al. "A Comprehensive Literature Review of Data Encryption Techniques in Cloud Computing and IoT Environment", ***International Conference on Information and Communication Technologies (ICICT)***, IEEE, Karachi, Pakistan,54-59 (2019).

16. Aljawarneh, S., M.B.J.M.T. Yassein, and Applications "A multithreaded programming approach for multimedia big data: encryption system", ***Multimedia Tools and Applications*** ,Springer Nature, Jordan,10997-11016 (2018).

17. Velan, P., et al. "A survey of methods for encrypted traffic classification and analysis", ***International Journal of Network Management***, Institute of Computer Science, Masaryk University, Brno, Czech Republic, Ponava,355-374 (2015).

18. Naralasetty, T. and K. Eswar "Secure Data Transmission Using Cloud Computing", ***INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)***, IJERT, Los Angeles ,USA,5 (2013).

19. Gai, K., et al. "Privacy-aware adaptive data encryption strategy of big data in cloud computing", ***IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)***, IEEE, Beijing, China,273-278 (2016).

20. Sekar, K. and M. Padmavathamma "Comparative study of encryption algorithm over big data in cloud systems", ***3rd International Conference on Computing for Sustainable Global Development (INDIACom)***, IEEE, New Delhi, India,1571-1574 (2016).

21. Zhang, Q. and Q. Ding "Digital image encryption based on advanced encryption standard (aes)", ***Fifth International Conference on***

Instrumentation and Measurement, Computer, Communication and Control (IMCCC), IEEE, Qinhuangdao, China,1218-1221 (2015).

22. Bensikaddour, E.-H., et al., "Embedded implementation of multispectral satellite image encryption using a chaos-based block cipher", ***Journal of King Saud University - Computer and Information Sciences***, 32 (1): 50-56.(2020).

23. Xu, Y., et al., "Design and implementation of distributed RSA algorithm based on Hadoop", ***Journal of Ambient Intelligence and Humanized Computing***, 11 (3): 1047-1053.(2020).

24. Reddy, K.R. and C.M. Rao "GUI implementation of image encryption and decryption using Open CV-Python script on secured TFTP protocol", ***AIP Conference Proceedings***, AIP Publishing LLC, Andhra Pradesh, India,020074 (2018).

25. Mahalakshmi, B., G. Deshmukh, and V. Murthy "Image Encryption Method Using Differential Expansion Technique, AES and RSA Algorithm", ***Fifth International Conference on Image Information Processing (ICIIP)***, IEEE, Shimla, India,363-366 (2019).

26. Subramanyan, B., V.M. Chhabria, and T.S. Babu "Image encryption based on AES key expansion", ***Second International Conference on Emerging Applications of Information Technology***, IEEE, Kolkata, India,217-220 (2011).

27. Al-Sawwa, J., et al., "Parallel particle swarm optimization classification algorithm variant implemented with Apache Spark", ***Concurrency and Computation: Practice and Experience*** 32 (2): 5451.(2020).

28. Terzi, D.S., R. Terzi, and S. Sagiroglu "A survey on security and privacy issues in big data", ***International Conference for Internet Technology and Secured Transactions (ICITST)***, IEEE, London, UK,202-207 (2015).

29. Mahajan, P., A.J.G.J.o.C.S. Sachdeva, and Technology "A study of encryption algorithms AES, DES and RSA for security", ***Global Journal of Computer Science and TechnologyNetwork, Web & Security*** Global Journals Inc, USA,9 (2013).

30. Padmavathi, B. and S.R.J.I. Kumari, India "A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution technique", ***International Journal of Science and Research (IJSR)***, R.V.S College of Arts & Science (Autonomous)Sulur, Coimbatore, India 5 (2013).

31. Alanazi, H., et al. "New comparative study between DES, 3DES and AES within nine factors", ***Journal of Computing***, A. A. Zaidan152-157 (2010).

32. Singh, S., "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security", ***International Journal of Computer Applications*** 67 (19): 33-38.(2013).

33. Ahmad, S., et al. "A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets", ***International Conference on Networking Systems and Security (NSysS)***, IEEE, Dhaka, Bangladesh,1-5 (2015).

34. Abdullah, A.J.C. and N. Security, "Advanced encryption standard (aes) algorithm to encrypt and decrypt data"; **https://www.researchgate.net/publication/317615794** (2017).

35. Abbood, F.H., A.A. Noor, and H.J.A.-M.J. Abed, "Proposal of New Block Cipher Algorithm Depend on Public Key Algorithms"; **https://doi.org/10.36541/0231-000-026-010** (2016).

36. Shaji, N. and P.J.P.T. Bonifus, "Design of AES architecture with area and speed tradeoff", ***Procedia Technology***, 24 (1): 1135-1140.(2016).

37. Hoang, V.-P., et al. "A low power AES-GCM authenticated encryption core in 65nm SOTB CMOS process", ***IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)***, IEEE, Boston, MA, USA,112-115 (2017).

38. Daemen, J. and V. Rijmen, "AES proposal: Rijndael", ***citeseerx***; **http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.640** (1999).

39. Cho, J., et al., "Power dissipation and area comparison of 512-bit and 1024-bit key AES", ***Computers & Mathematics with Applications***, 65 (9): 1378-1383.(2013).

40. Oh, J.-Y., D.-I. Yang, and K.-H.J.H.i.r. Chon, "A selective encryption algorithm based on AES for medical information", ***Healthcare Informatics Research***, 16 (1): 22-29.(2010).

41. Srisakthi, S. and A. Shanthi "Towards the Design of a Stronger AES: AES with Key Dependent Shift Rows (KDSR)", ***Wireless Personal Communications*** Springer Science+Business Media, LLC, part of Springer Nature, India,3003–3015 (2020).

42. Kocabas, O., et al., "Emerging security mechanisms for medical cyber physical systems", ***IEEE/ACM Transactions on Computational Biology and Bioinformatics***, 13 (3): 401-416.(2016).

43. Alam, G.M., et al., "Using the features of mosaic image and AES cryptosystem to implement an extremely high rate and high secure data hidden: Analytical study", ***Scientific Research and Essays***, 5 (21): 3254-3260.(2010).

44. Nechvatal, J., et al., “Report on the development of the Advanced Encryption Standard (AES)”, ***Journal of Research of the National Institute of Standards and Technology***, 106 (3): 511.(2001).

45. Le, D., et al. “Parallel AES algorithm for fast data encryption on GPU”, ***International Conference on Computer Engineering and Technology***, IEEE, Chengdu, China,V6-1-V6-6 (2010).

46. Elbirt, A., et al., “An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists”, 9 (4): 545 - 557.(2000).

47. Dworkin, M.J.J.o.R.o.t.N.I.o.S. and Technology, “SECOND ADVANCED ENCRYPTION STANDARD CANDIDATE CONFERENCE Rome, Italy”, ***Journal of Research of the National Institute of Standards and Technology*** 104 (4): 401-410.(1999).

48. Sybrandy, C. and J. Macdonald, “Public Comments Regarding The Advanced Encryption Standard (AES) Development Effort Round 2 Comments”, ***response to a notice***, 64 (178): 50058-50061.(1999).

49. Coppersmith, D., et al., “A proposed mode for triple-DES encryption”, ***IBM Journal of Research and Development***, 40 (2): 253-262.(1996).

50. Cahya, R., et al., “Noble Method for Data Hiding using Steganography Discrete Wavelet Transformation and Cryptography Triple Data Encryption Standard: DES”, ***International Journal of Advanced Computer Science and Applications(IJACSA)***, 9 (11): 261-266.(2018).

51. Coppersmith, D.J.I.j.o.r. and development, “The Data Encryption Standard (DES) and its strength against attacks”, ***IBM journal of research and development***, 38 (3): 243-250.(1994).

52. Sklavos, N., G. Selimis, and O. Koufopavlou “Bulk encryption crypto-processor for smart cards: design and implementation”, ***Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems.***, IEEE, Tel Aviv, Israel,579-582 (2004).

53. Aleisa, N.J.I.J.o.S. and I. Applications, “A Comparison of the 3DES and AES Encryption Standards”, ***International Journal of Security and Its Applications***, 9 (7): 241-246.(2015).

54. Karau, H., et al., “Learning spark: lightning-fast big data analysis”; **http://oreilly.com/catalog/errata.csp?isbn=9781449358624** (2015).

55. Zaharia, M., “An architecture for fast and general data processing on large clusters”; **https://doi.org/10.1145/2886107** (2016).

56. Armbrust, M., et al. “Spark sql: Relational data processing in spark”, ***ACM SIGMOD international conference on management of data***, Association for Computing Machinery, New York, NY, United States,1383-1394 (2015).

57. Zaharia, M., et al. “Discretized streams: Fault-tolerant streaming computation at scale”, ***Proceedings of the twenty-fourth ACM symposium on operating systems principles***, Association for Computing Machinery, New York, NY, United States,423-438 (2013).

58. Meng, X., et al., “Mllib: Machine learning in apache spark”, ***Journal of Machine Learning Research***, 17 (1): 1235-1241.(2016).

59. Gonzalez, J.E. “From graphs to tables the design of scalable systems for graph analytics”, ***Proceedings of the 23rd International Conference on World Wide Web***, UC Berkeley, CA, USA,1149-1150 (2014).

60. Salloum, S., et al., “Big data analytics on Apache Spark”, ***International Journal of Data Science and Analytics***, 1 (3-4): 145-164.(2016).

61. Zaharia, M., et al., “Spark: Cluster computing with working sets”, ***HotCloud'10: Proceedings of the 2nd USENIX conference on Hot topics in cloud computing***, 10 (10-10): 95.(2010).

62. Mavridis, I., H.J.J.o.S. Karatza, and Software, “Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark”, ***Journal of Systems and Software***, 125 (1): 133-151.(2017).

63. Arfat, Y., et al., “Parallel shortest path big data graph computations of US road network using apache spark: survey, architecture, and evaluation”, ***Smart Infrastructure and Applications***; **https://link.springer.com/chapter/10.1007/978-3-030-13705-2_8** (2020).

64. Coulon, C., E. Pacitti, and P. Valduriez “Consistency management for partial replication in a high performance database cluster”, ***International Conference on Parallel and Distributed Systems (ICPADS'05)***, IEEE, Fukuoka, Japan,809-815 (2005).

65. Zhao, Y., et al. “Big data processing with probabilistic latent semantic analysis on MapReduce”, ***International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery***, IEEE, Shanghai, China,162-166 (2014).

RESUME

Muhammed Yaseen ALHAYANI was born in Iraq/Mosul in 1994 and he graduated first and elementary education from Mosul city. He completed high school education in Omer bin Abd Alaziz High School, after that, he started undergraduate program in Mosul University Department of Computer Scince in 2012. Then in 2019, he started assignment as a Research Assistant in Karabuk University Department of Computer Engineering. To complete M. Sc. education, he moved to Karabük University, where he has been still working as a R. A. for

CONTACT INFORMATION

Address: Karabük University
Department of Computer Engineering
Öğretmenler sitesi /KARABUK

E-mail: malhayani94@gmail.com ; stones1994aa@gmail.com