Chapter One

1. Introduction

a. Background

Despite the fact that electronic commerce is not commonly used digital platform in Ethiopia to exchange products and services via the internet between producers and consumers of products and services in the global market due to a variety of factors including infrastructure, strong financial institutions, and knowledge. E-commerce is a simple and effective system via which purchasers may offer their products and services to a global market and generate cash by selling products and services to the potential buyers.

Business-to-Business e-commerce transaction is the most popular and also a business type which generate huge amount of money than all the other types of e-commerce. [1]

Various business firms are exchanging very sensitive information's, money and also sample pictures of various products to be sold online in terms of textual presentation, Video display, image formats and sound/Audio explanations.

E-commerce (electronic commerce) is a web-based system that orders a product or adds it to a basket and settles the transaction using online payment methods such as credit cards and debit cards once the buyers agree to buy it. The most common types of e-commerce transactions are Business-to-Business (B2B), in which businesses exchange goods and services among themselves, Business-to-Consumer (B2C), in which enterprises sell their products and services to customers online, and Consumer-to-Consumer (C2C), in which consumers sell their products to other shoppers. [1]

In addition electronic commerce has came up with a different mechanism to approach potential buyers through different technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. The technologies used will support organizations on increasing sales, business efficiency and provide a basis for new products and services in a global market. [2]

On transferring money from the buyers account to the sellers, the communication channel usually being intercepted by hackers and crackers. Hence security in e-

commerce is a fundamental concept which deals with the protection of all the property of the web from unauthorized access, use, alteration, and destruction.

Integrity (ensuring that any information that customers have shared online remains unaltered), Non-repudiation (a legal principle that instructs players not to deny their actions in a transaction), and Truthfulness (both the merchant and the purchaser should be real) are the basic dimensions of e-commerce security. They should be who they say they are), and Safety (avoiding any conduct that would result in the sharing of consumers' data with uninvited third party companies). [3]

Different security ensuring mechanisms are introduced and implemented in different websites on their digital payment mechanisms among the most popular are:the Deffie-Hellman Algorithm, Message Digest Algorithm 5 (MD5), Advanced Encryption Standard (AES), Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Ravish Shamir Algorithm are some of the most popular security ensuring mechanisms introduced and implemented in various websites on their digital payment mechanisms (RSA). [3]

This paper measures the performance of Advanced Encryption Standard algorithms and enhancing the strength of the security and the performance of the algorithm among the above listed cryptographic algorithms to facilitate the encrypt/decrypt procedures of data blocks of different files size through experimentation and measure the changes in terms the file size and data types using a python program. Since python is among the modern programming languages which is rich in terms of content. The modified AES algorithm can also be combined with the other key management standard the Message Digest 5 (MD5) hashing algorithm to secure the key exchange in an electronic business from the buyer to the seller. .

b. Problem Statement

Hackers and crackers are snatching, damaging, and destroying the property of individuals and organizations, posing a significant threat to worldwide electronic commercial activities. New tech is also helping them by providing improved computer platforms that support them to crack huge block size algorithms; as a result, numerous commercial sites use algorithms such as The pace of transactions between the buyer and the seller is slowed by AES and these algorithms. Even if various researchers in the field have proposed various ways to assure optimum security, there

is still a delay in transaction execution time and a loss of confidentiality within corporate enterprises. [1]

Among such business firms complaining due to the sluggish execution time on file transfer. Among the factors considered on reducing the execution time is the structure of the algorithm of AES standard. [6]

AES is a specific key encryption algorithm used for both encryption and decryption. Since 2001 G.C., the standard AES with 128 bits has used 10 rounds of procedure. However, there have been no substantial attacks that have breached the security for about 20 years, so spending much more time on executing all 10 rounds will not be efficient. As a result, the main focus of this thesis is on altering the AES algorithm and combining it with MD5 for secure key exchange [20].

There are a few researches made on identify the best combinations of cryptographic algorithm to enhance the performance and to ensure maximum security specifically on B2B ecommerce activities and also as papers that I have reviewed, there is no paper dealt on combining AES encryption with MD5 hashing algorithm to get a benefit of ensuring maximum security of the encryption process.

Research Questions

- What are the factors which needs to be consider to modify AES Cryptographic algorithm?
- What are the problems associated with B2B e-commerce transactions on exchanging of goods and services online?
- Which cryptographic algorithms can be combined with AES standard to enhance the performance and security of B-to-B e-commerce transactions?

c. Objectives

General Objective

The general objective of this thesis is to modifying AES algorithm and to combine it with the MD5 hashing algorithm to improve the performance and strength of ecommerce security.

Specific Objectives

The specific objectives to be achieved by this thesis are:

- To measure the performances of the selected cryptographic algorithm and measure its response time on different files with different size and data type.
- To identify the essential properties of the MD5 hashing algorithm.
- To modify the Advanced Encryption Standard (AES) by increasing its security and reducing its computation time through reducing the number of rounds from 10 to 8.
- To combine the modified AES algorithm with the MD5 hashing algorithm to ensure better security and performance of the electronic commerce key exchange mechanism.

d. Significance of the Study

The beneficiaries of this research thesis are various business firms engaged on Import and export companies whom are buying and selling of goods and services online through e-commerce and third parties like financial institutes which are intermediaries between buyers and sellers.

It also create awareness for educational sectors and scholars on the area on how the cryptographic algorithms can be modified for better performance and maximum security. More over governments can also get the benefits from the smooth transactions in the global market and attract foreign investment that will positively contribute to the economy.

e. Scope and limitation of the Study

The study is limited to evaluate and measure only the performance of the AES algorithm and modify it to reduce the response time of encrypting and decrypting of business transactions and messages to be transmitted on to an electronic commerce. This thesis also focuses on combining the AES algorithm with that of the message digest 5 hashing algorithm to ensure the security of key exchange between the sender and receiver.

f. Organization of the study

This research paper is organized by five different chapters and the first chapter deals with Introduction part which deals about the general view of the paper, chapter two deals with literature review which summarizes the researches done by different scholars in this specific topic of research and chapter three specifies the methods, data and techniques employed on this paper to study and analyze the findings and to present. Chapter four also deals with the experiments, results and findings of the study in a summarized way and finally chapter five will come up with the final conclusion and summary of the final results and will indicate the future work to be done in line with this specific research topic.

Chapter Two

Literature Review

2.1. Introduction

Electronic commerce is defined by Martin [1] in his book as a computer-based market system in which vendors advertise their products and services to potential buyers and customers look for a product or service to acquire. There are many famous websites created to connect consumers and businesses via the internet and a web-based personal computer, which then offers a payment mechanism based on the purchasers' and sellers' consent. A payment will be maintained and money will be released using forms of electronic communication.

2.2. Ecommerce security

In his book, Martin [1] considered as a reliable state in which a person, a resource, or a procedure is safe from a harm or its harmful acts. The security of our information asset is referred to as information security.

It also has generally referred to as “information security protection goals, which are as follows: Authenticity is considered as the provable truth or believability of an object or subject. Integrity as data cannot be modified without being discovered and without proper authorization. Confidentiality: retrieving of information is impossible without appropriate permission. Availability: Subjects who have been authenticated and approved will not be restricted in their rights until they have received necessary authorization. Authorization is the power and right to carry out a task.

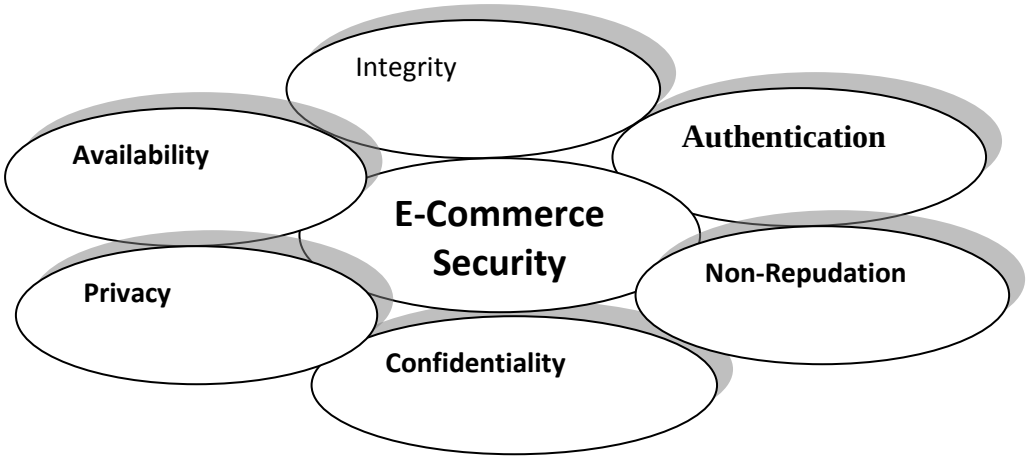

Figure1. Dimensions of Ecommerce Security

According to him, E-Commerce enterprises use online stores for direct-to-purchase retail sales, offering or engage in online marketplaces that handle third-party producers-to-consumer or end user to end user sales, producer-to-producer electronic data exchange, marketing to potential and existing customers via email or fax, and engaging in or before for the launch of new goods and services.

E-commerce is explained in the book as the buying and selling of products between independent bodies and or individuals, supported by the ubiquitous use of powerful Computer systems and globally standardized technology deployment. The book also mentions some benefits of e-commerce for customers, such as adaptability in weekly shop hours, no forced to wait lines once the network is fully operational, the willingness to shop at home, personal needs can be met through personalization, international offers, more competitive pressure, price pressure, and improved customer service from providers, accelerated communication with customers, international visibility, and no protectionist measures.

It has also mentioned some disadvantages like Security threats linked with data breaches (credit card stealing), impersonation (name or user account), abuse, offenses such as fraudulent firm (firm that does not exist), deception (invoice is paid, but items are never delivered), and ambiguous legal status are all discussed.

The book at the end has also defined security as "security is a status where a person, a resource or a process is protected against a threat or its negative consequences.

Communication security refers to the protection of our digital content. Finally, the book mentions the adoption of cryptographic techniques as an effective cyber security strategy for e-commerce.

According to Ann and Murphy et al [2], "The Role of Cryptography in Security for Electronic Commerce," have indicated that digital economy encounters sets of problems and possible dangers, including unauthorized entities, embezzlement, tampering, and damage of all the organizational information systems. The article examines major security concerns of enterprises' and users', and also the cryptographic algorithms used to mitigate potential threats.

The paper also deals with the key security issues of organizations and consumers interacting in electronic commerce, with a focus on the role of cryptology in lowering the danger of doing business over the internet. Cryptologists' approaches and nomenclature, such as private and public key cryptography, as well as their application and benefits in safeguarding both information and information systems, are explained in depth.

The problem of biometric authentication verification and verification agencies is examined.

According to them security, Secured Socket Layer (SSL) ensures safe connection between the client and the server by data encryption and authenticating the server using public advanced technologies. SET improves SSL by offering ways to secure User Information and the security of monetary operations through the segregation of merchant and payment information, despite being more complex and expensive to implement. Finally, they advise that online security will always be a never-ending challenge for companies who want to be confident that their digital payments are safe.

As improvements in protocols, hacking tactics will improve when protocols, identification, stability, security systems, and secrecy increase. Humans will continue to be in charge of constantly monitoring digital infrastructure, as well as assessing and deploying hardware and software solutions.

As properly stated by Churi [3] the Secure Electronic Transaction (SET) protocol for processing transactions on ecommerce. SET is a comprehensive security protocol that employs cryptography to assure information secrecy, payment integrity, and identity

authentication. To provide communication secrecy and security, it uses encryption, a digital certificate, and SMS verification. The paper begins by discussing online store and how to build them. It then goes on to describe how SET works and the various parts that make it up. The report then details the protocol's formulation and construction.

The paper also argues that Consumer and retailers will have a similar set of demands and concerns when it comes to internet commercial transactions: Security: Electronic currency is nothing more than data that may be readily stolen. It must be verified that no one else can reroute a payment or impersonate somebody to take his money. Furthermore, each partner should be safeguarded from collaboration by others (multi-party security). To protect his safety, no participant in the organization needs to trust another participant — or at least as little trust as practicable. The acclaimed security qualities must be provable by the general public. The intention was to actually implement the SET standard by addressing its drawbacks and integrating equal capabilities to provide a superior safety mechanism.

Consumer verification was achieved by implementing a verification code to the customer's phone, and website authentication was accomplished by using a standard SSL certificate.

Powerful cryptographic algorithms are used to keep the customer's confidential data, such as encrypted messages and credentials, in scrambled form.

In their essay, Niranjanamurthy and Dharmendra [4] attempted to grasp the fundamentals of ecommerce security, which they define as the preservation of e-commerce properties from security breaches, use, modifications, or annihilation. Fidelity, Non-repudiation, Validity, Secrecy, Privacy, and Accessibility are the aspects of e-commerce cyber security. E-commerce provides great possibilities for the financial sector, but it also introduces new risks and vulnerabilities, such as security concerns. As a result, information security is a critical managerial and technical need for any efficient and effective online payments transaction activities. Nonetheless, due to ongoing technology and market development, its development is a difficult undertaking that requires a synchronized match of algorithm and practical solutions. They explored the key areas in their paper: Overview of E-commerce protection, Understanding Internet Shopping Steps to Place an Order, Purpose of E-commerce Confidentiality, Different E-commerce Security Concerns, and Protecting Virtual Shopping Guidelines.

The authors have used common guidelines to study the common problems and concluded that Common mistakes that leave people vulnerable include purchasing online that aren't secure, giving out too much private information, and leaving computers open to malwares.

They also talked about e-commerce security concerns, security procedures, the digital e-commerce cycle/online purchasing, security risks, and suggestions for convenient and secure retail on shopping sites.

The limitation of this paper is that it does not mention the appropriate methodology used to draw the required outcome of the paper and hence did not add any new value or there is no new finding.

Ejub Kajan [24] specializes in security issues regarded with a company-to-company electronic commerce security and violence. B2B is a revolutionary way to conduct business through the Internet. The B2B science community is interested primarily in ad - hoc basis interconnection between commercial enterprises. This study tries to figure out where security fits into the big picture of B2B device-to device exchanges.

The study also found that the security standards for Payments are extremely stringent. Business data must be protected before, after, and during transactions on the transmission line. Many times, a financial transaction takes place between two participants who do not trust each other, and when a third party is engaged in the process.

The issue of trust is a major concern in B2B cyber security. As a result, we must define precisely what constitutes a safe (trusted) personal computer (network).

2.3. Cryptographic algorithm

Cryptology is a mathematical discipline that includes both cryptography and cryptanalysis. Cryptologists today typically have a background in speculative mathematics and computer science. Cryptography is a method and technology of maintaining communications safe, and cryptographers perform it. Cryptanalysts are experts in cryptanalysis, which is the application of science of deciphering cipher text, or seeing plaintext behind the cryptographic mask. Cryptography has a unique challenge that is not encountered in other areas of study: the requirement for daily communication between cryptography and cryptanalysis.

This interaction begins with a request from security experts, who begin each cycle by introducing a new optimization technique and receiving a response from cryptanalysts, who attempt to disclose weaknesses in the model, which is frequently significantly more difficult than inventing the algorithms in the first place. As a result of this healthy competition, "strong" cryptography emerges.

Cryptography is crucial to network security, particularly in the process of exchanging and transmitting data through open channels. Cryptography is required in order to protect the data.

Asmaa Essam Alhibshi [5] focuses on evaluating the performance of a few different encryption protocols. It denotes the importance of cybersecurity in data protection. The paper provides an overview of cryptography. This research analyzes the most widely used cryptographic techniques, RSA and AES, and assesses their effectiveness in Java and C++.

It begins with a quick overview of the many forms of cybersecurity, such as symmetric and asymmetric cryptographic protocols. The past work on these encryption algorithms is then discussed, along with an assessment of their performance. Encryption techniques are divided into two categories in the paper: symmetric and asymmetric algorithms.

Both the encryption process and decryption process keys are the same in symmetric key encryption techniques. The transmitter and receiver will both were using the same key to encrypt and decrypt, which is known as symmetric encryption. Symmetric-encryption techniques include TRIPLE DES, BLOWFISH, AES, RC4, and RC6.

Asymmetric key-encryption techniques: The key used to encrypt and decrypt information will be unique. Examples include RSA, DSA, and Diffie-Hellman.

The researcher has identified that Cryptography, according to the experts, plays a vital role in computer security, carried out with the intention of exchanging and transmitting information through open channels. Cryptography is required in order to protect the data.

Finally, the researcher employed standard Java versions of the AES and RSA algorithms, while writing his own encryption algorithm in C++. All packets are received in 0.17035 seconds in Java (AES) and in 0.17035 seconds in Java (RSA). 1.9268 seconds. In C++ (AES) all packets are received in 1.388925 seconds and in C++ (RSA) all packets are received in 2.954425 seconds. Hence concluding that, AES and RSA both perform better in Java.

Packet speed was determined by the average time. Because JAVA is a more advanced language as compared to that of C++, the model under test ran faster. JAVA programs operate underneath the Java Runtime Environment (JRE), which automates trash collection and prevents memory leakage, which we have to perform explicitly with C++.

According to the study, it was difficult to write the encryption technique in C++ because there was little guidance available for C++ in such a situation. Also, as previously stated, in C++, we must delete data variables manually erase data variables to reduce resource leakage, but in JAVA, this is automatically created. Moreover, whereas in JAVA there are well-defined types of data to employ in programming, but in C++ there are no well-defined data types, making programming in C++ difficult.

2.4. DES, 3DES, Blowfish and AES (Rijndael) encryption decryption algorithms

Abdel-Karim Al Tamimi [6] discusses the two primary criteria that distinguish one encryption method from another: the ability to safeguard the protected data from attacks, as well as the speed and efficiency with which it does so. This paper compares the performance of four of the most widely used encryption algorithms: DES, 3DES, Blowfish, and AES (Rijndael). The evaluation of the algorithm's encryption/decryption speed was done by running numerous encryption settings to handle varying sizes of data blocks. Then the algorithms were evaluated in terms of the time necessary to encrypt and decrypt data blocks of various sizes (0.5MB to 20MB). All of the implementations were meticulous in order to ensure that the outcomes would be reliable and objective.The results show the superiority of

Blowfish algorithm over other algorithms in terms of the processing time. It shows also that when the data block size is rather large, AES consumes more resources, and 3DES always takes longer than DES due to its triple phase encryption feature. Despite having a lengthy key (448 bits), Blowfish outperformed competing encryption techniques. Worm holes have been discovered in the security mechanisms of DES and 3DES, but none have been discovered in Blowfish or AES so far. As the results show, Blowfish takes the least amount of time to decrypt a 20 MB data file (approximately 3 seconds), followed by DES (4.5 seconds per 20 MB), 3DES (6 seconds per 20 MB), and AES (16 seconds).

The strength of this paper is use of simulation technique for measuring encryption and decryption on the web. The limitation is that the paper did not add value or does not come with a new finding it simply measure the performance and compare with the other algorithms and select the algorithm with least time.

"Performance Analysis on the Implementation of Data Encryption Algorithms Used in Network Security," according to Abraham Lemma [7], claims that E-commerce is unreliable without security. Security is not as straightforward as it may look to the untrained eye. The criteria appear to be simple; indeed, most of the primary security service needs may be summarized in a single word: secrecy, authentication, non-repudiation, or integrity. In the realm of network security, cryptography is critical. Many encryption techniques are already available to safeguard data, but they use a significant amount of computational resources, such as memory and CPU time. The major topic of this study is a comparison of four symmetric encryption algorithms: DES, Triple DES, AES, and Blowfish.

The performance of these methods is compared and assessed using encryption and decryption duration, performance, and memory consumption. On two computers with different operating systems, the developments are performed out using the Java software. The author used Net Beans IDE 7.4 to assess the performance of encryption and decryption algorithms simulated with JDK jdk1.7.0 45. To test the performance of encryption techniques, several simulations were run with varied text sizes. Finally, the paper compares four cryptographic algorithms: DES, AES, Triple DES, and Blowfish, which are incorporated in the strong handheld programming language Java and JCA (Java Cryptography Architecture), which are used in implementing the encryption algorithms, in various scenarios with various file sizes.

The findings are compared and required conclusions are drawn to assess the performance of four cryptographic methods. The study came to a conclusion based on the performance of cryptographic algorithms as measured by a metric.

According to the study, Triple DES takes longer to encrypt/decrypt, uses less memory, and has a low throughput. AES and Blowfish both take the same amount of time to encrypt/decrypt and have higher throughput, but AES requires more memory than Blowfish. DES also requires the same amount of memory as Triple DES, but it takes the least amount of time to encrypt/decrypt and has a better bandwidth.

The thesis strength stems from the application of proper techniques and approaches to assess the algorithms' effectiveness. As a flaw, the author failed to identify a creative idea or bring new value to the article; instead, the same output produced by other researchers was duplicated.

H. Arif and S. Zarina, et al. [8] are working on a fast and secure electronic payment system for e-commerce so that customers can connect with the vendor right away. Even if the client can disguise his or her identity and create a temporary identity to execute the service, the proposed system does not require the customer to submit his or her identity into the merchant's website. The authors' protocol has been determined to have significantly increased security performance in terms of secrecy, consistency, non-repudiation, secrecy, identification, and permission.

Client (C), retailer (M), payment gateway (PG), user bank (B), and financial institution are the five entities that make up a conceptual system.

They perform in the following manner. Each entity, such as the client, seller, user banks, and merchant bank, must register with the payment gateway in order to create a secret key. Secure c onnectivity necessitates the use of secret key elements. In addition, the user and the merchant construct a secret key between them. The purchaser investigates the seller and orders the product, now that he or she has formed a temporary identity on the merchant's website, and the retailer submits the request to the payment gateway.

The paper compared the proposed architecture to the other three current methods, which use RSA and DES to encrypt and anonymize debit/credit card information.

The majority of clients prefer an e-commerce program because it offers numerous benefits. Clients require such a security mechanism since it meets all requirements and is adequate.

On the basis of these requirements, the researchers presented a secure digital currency for e-commerce contexts. They developed a mechanism in which the payment gateway acts as a go-between for the client/merchant and the bank. The suggested technique has higher protection efficacy in terms of confidentiality, non-repudiation, integrity, availability, and anonymity, according to the security analysis.

The strength of this work is that it introduces a novel mechanism known as the payment protocol for e-commerce; nevertheless, its implementation and performance have not been studied.

The paper "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA, and Blowfish" by Priyadarshini, Prashant, and others [9] discusses the different cryptographic techniques that can be used to ensure ecommerce confidentiality and argues that a user needs a cryptographic techniques that is low cost and high performance. However, in fact, such a one-stop solution algorithm does not exist. A number of algorithms have a cost-performance trade-off. A financial application, for example, necessitates maximum security at a significant cost, whereas a gaming implementation sending player patterns for analyses does not care about security and must be quick and cost-effective.

The researchers have determined which of the existing cryptographic algorithms best meets the user's needs, and a study of each algorithm's advantages, weaknesses, cost, and performance will reveal significant information. The researchers then developed and studied in detail the performance and efficiency of widely used cryptographic algorithms DES, 3DES, AES, RSA, and blowfish in their article to demonstrate a current performance analysis.

DES, 3DES, AES, blowfish, and RSA have all been developed and compared by the researchers. The Java Eclipse IDE was used to create an algorithm. They used the java security and java crypto packages. They employed java crypto and security modules, which provide security features such as encryption, decryption, key generation, identity management architecture, authentication, and authorization. Blowfish, on the other hand, isn't included in the Java security and crypto libraries.

They've also written blowfish in Java, turned it to a jar, and added the jar to the crypto library's external library. They employed text and image files with sizes of 25KB, 50KB, 1MB, and 2MB, 3MB as input for encryption. Each file's encrypted output is saved in a file, which is then used to decode it. Throughout the experiment, they can use the same input files for all algorithms for the sake of comparison.

They used the same system for all implementations and analyses so that memory and processor conditions were consistent across all algorithms. All block cipher algorithms are set to the same mode, which is ECB, which is the default in Java cryptography and security.

Finally, the results demonstrate that RSA takes the longest time to encrypt data, whereas blowfish takes the shortest time to encrypt data, making it the fastest. 3DES is a technique for reusing DES implementations that involves cascading three DES instances with distinct keys. 3DES is thought to be safe. The result also shows that RSA takes the most time to decrypt, while blowfish takes the least time and is the fastest. The researchers also stated that RSA is slower than symmetric key algorithms due to the use of modular multiplication, multiplicative inverse, and two keys, public and private. The result also shows that Blowfish uses the least amount of memory per unit of processing, whereas RSA uses the most. The RAM requirements for DEA and AES are moderate.

The paper's strength is the suitable technique of testing with the results and their presentation; nevertheless, the integration on the web was not done using an adequate simulation.

On [10], Norman D. Jorstad focuses on the development of measures for describing the strength of cryptographic methods. In this study, we look at a small number of symmetric cipher block encryption techniques in the codebook mode, as well as an asymmetric public key algorithm. Other communication consistency, verification, and biometric authentication methods and cryptographic techniques were not studied. For his research, he relied solely on publicly available information.

Six symmetric or secret key (one-key) block ciphers (DES (DEA, 3DES (EDE), SKIPJACK, RC4TM, RC5TM) and one asymmetric or public key (two-key) algorithm (RSA) were actually selected as an approach to give a modest representative selection of well-known cryptographic algorithms.

The authors feel that this tiny sample population of cryptographic algorithms, as well as examples of the types of metrics that could be used to quantify algorithm effectiveness, is adequate to demonstrate that cryptographic algorithms and associated systems could be valued. They might also be used for defining Common Criteria assurance levels for cryptographic subsystems or operational departments. These metrics could have been useful for assessing and comparing service offerings, albeit an alternative (or extra) set of scales for rating the strength of cryptographic functionality in distinct goods could be devised.

The merit of this research is that it investigates eight methods utilizing the factoring algorithm and generates some pilot metrics for the aforementioned cryptographic techniques. The paper's flaw could be that it failed to mention the simulator and software used to test the experiment and determine the outcome.

2.5. Message Digest 5 Algorithm

Priyanka Walia [11] works on expanding the hash code length to 512, which makes the technique more resistant to collisions. A hash function computes a set length output termed the message digest from an input message of varied lengths, according to the author. The MD5 message digest method was created by Ron Rivest at MIT and produces a 128-bit hash code from a message input of various lengths. It's one of the most popular hashing algorithms. However, it has been reported that the algorithm poses a security risk.

Furthermore, in the near future, a 128-bit hash result may not provide sufficient security protection. Finally, the researchers have built a unified design for MD5 and MD5-512 by working with the Message-digest Algorithm 5.

Hanna Willa et al. [12] concentrate on the two basic cryptography procedures of encrypting and decrypting with the same key. Secret Key, Shared Key, or Symmetric Key Cryptosystems employ the same key for both encryption and decryption processes.

The encryption procedure makes the message readable (plaintext) but randomly unintelligible (cipher text), while the decryption process is the inverse of encryption, converting the cipher text plaintext using the same key. We'll employ password-based encryption schemes in the encryption and decryption processes to perform symbolic

and operational cryptography using symmetric, asymmetric, and password-based encryption.

The researcher created a program that, depending on which button is pressed, will generate an encryption and decryption file. To enter the default Iteration hash password 100, the program uses PBE with MD5. The procedure generates the secret key of an artificial password named Password-Based Encryption and by applying MD5 because it uses an algorithm that combines standard hashing and encryption methods as well as DES that works in plain text useful for going back the same size cipher text.

Douglas Selent [13] the researcher was tried to create an algorithm that was resistant against known attacks, simple, and quick to code by choosing field GF(2)8. Exclusive-OR operations (XOR), octet substitution with an S-box, row and column rotations, and a Mix Column are all used in the algorithm. According to him, AES was originally created for non-classified US official statistics, but because to its success, AES-256 can now be used for top-secret government data. As of July 2009, no real attacks on AES has succeeded.

The paper's strength is that it maintains a new technique to change the AES encryption standard, while its drawback is that it does not use a simulator to assess performance from the transmitter to the receiver.

Anurag Rawa, et al. [14] describe several algorithms for assuring safer information transfer, ranging from classical ciphers to today's hash function. Cryptography, according to the writers, allows data or information to be transmitted via a network in an unidentifiable manner so that intruders cannot decipher it.

Only the source and intended destination can read and comprehend the message due to cryptography's methodology. Starting with letter substitution and progressing to today's unbreakable public key cryptosystem, cryptography has progressed over time.

They also discovered how cryptography has changed through time, from centuries ago when cryptography techniques were mostly utilized for army and parliamentary goals to now, when these algorithms are primarily used for data security and cryptographically secure.

Finally, they discovered that the BLOWFISH Algorithm is the most secure of all symmetric cryptography. Many studies have shown that the BLOWFISH algorithm outperforms all other block ciphers in terms of efficiency. The next technique used to protect our data is RSA, which is currently employed in a large number of applications.

They also suggest that the RSA method be combined with additional algorithms like as RSA and DES, RSA and AES, RSA and DIffie Hellman, and RSA and IDEA to make the information more secure.

As limitation the researcher did not came up with a novel idea they only repeat and recommend what is being done by other researchers too.

The main focus of Abraham Lemma [15] is on a comparison of four symmetric encryption algorithms: DES, Triple DES, AES, and Blowfish. The effectiveness of these methods is compared and assessed using encryption and decryption time, throughput, and memory utilization. On two PCs with distinct operating systems, namely Windows 7 and Windows 8, the operations are carried out using the Java software.

Triple DES took longer to encrypt/decrypt, used lesser memory, and had a low throughput, according to the results. AES and Blowfish both take the same amount of time to encrypt/decrypt and have higher throughput, but AES requires more memory than Blowfish. DES also requires the same amount of memory as Triple DES, but it takes the least amount of time to encrypt/decrypt and has a higher throughput. According to the results, the Blowfish encryption/decryption algorithm performs better than the other algorithms. For files larger than 2547kb, the outcome may be changed.

2.6. Hybrid cryptographic algorithms

Abdul Monem [16] proposes a cipher method that improves Diffie-Hellman key exchange by combining the MD5 hashing algorithm, AES symmetric key algorithm, and Modification of Diffie-Hellman asymmetric key algorithm in a discrete logarithm problem (DLP) to increase the complexity of this method over an unsecured channel (MDH).

The researchers have used Irreducible truncated polynomial mathematics to build the proposed system because it is highly efficient and compatible with personal computers.

They incorporate the best aspects of both symmetric and asymmetric encryption. The AES algorithm is used to encrypt the data (plain text) that will be transferred. To generate an AES key, the data (plain text) was fed into MD5. This key was encrypted using a DiffieHellman modification (MDF). The MD5 technique is important in two

ways: first, it guarantees the integrity of the data being transferred, and second, it makes it simple to generate the secret key that is used in the AES process.

The results reveal that for a 1000 character message, AES takes about 0.30 milliseconds to encrypt, 0.17 milliseconds to decrypt, while MDH takes roughly 0.700 milliseconds to encrypt, 0.500 milliseconds to decrypt, and MD5 takes 0.20 milliseconds.

They came to the conclusion that using a hybrid cryptographic techniques will almost certainly improve the performance of cryptographic algorithms. Confidentiality, integrity, and authentication will all be guaranteed by this protocol. The AES method ensures confidentiality, the MD5 hash function ensures integrity, and DiffieHellman modification ensures verification.

We've put the algorithm through its paces with a variety of message sizes. The model increased interactive performance while delivering high-quality security service for intended e-commerce operations, according to the trial results. The use of hybrid models by modifying DiffieHellman is a strength, but the tool used to simulate the test is not mentioned.

Prakash Kuppuswamy and Saeed devised a hybrid cryptography system that combines the symmetric key method and the widely used RSA algorithm. In all data security purposes, the symmetric key algorithm based on integer numbers and the RSA method are frequently used. The efficiency of security approaches is elevated, and such competence grows as security methods are coupled.

The researchers looked at the performance of the existing DES, 3DES, and AES algorithms to come up with a novel hybrid power system.

The programs written in MATLAB and their performance tested in real-world scenarios are saved in three distinct arrays for key generation, encryption, and decryption schemes. It's been put to the test with a length of 100 bits.

The result shows that the DES algorithm with a key size of 64 bits is used. The AES method with a key size of 256 bits took 10 seconds to encrypt and decrypt. 3-DES with a key size of 2112 took 20 seconds to encrypt and decrypt. The SSK+RiSA with key size of 2,048 had a 15-second encryption/decryption time, while the SSK+RiSA with key size of 2,048 had a 10-second encryption/decryption time. The paper's drawback is that it solely calculates algorithm performance without evaluating text transmission between the client and server simulation machines.

Sidraah Matte, et al. [18] experimented with several cipher algorithms that improve the DiffieHellman key exchange by employing reduced polynomial in discrete logarithm problem (DLP) that boosts the safety of internet-based e-commerce transactions. It also includes algorithms like MD5 and AES. MD5 is an asymmetric key method, whereas AES is a symmetric key algorithm.

Experimental analysis was utilized to improve the DiffieHellman key exchange utilizing a truncated polynomial in a discrete logarithmic issue in order to raise the method's complexity over an insecure channel.

The paper's flaw is that no specific results are presented for boosting the effectiveness of the novel hybrid method. It makes no mention of the simulator that was used to test the algorithm on both the client and server sides.

Jothina Mazarir and Clopas Kwenda [19] dealt with the AES algorithm, which is more widely used for securing ecommerce transactions than all others, and began to investigate the major limitations associated with the algorithm's performance on running a transaction, namely that the algorithm is slow in both the encryption and decryption processes, and thus could not handle large volumes of data. With this dilemma in mind, a hybrid algorithm was created by combining AES and RSA in the hopes that RSA's flaws would be mitigated by AES and vice versa.

This study by Edjie M. De Los Reyes [21] focuses on lowering the number of rounds in the AES algorithm from ten to six in order to ensure file secrecy. The AES encryption round was modified by reducing the round repetitions from 10 to 6, adding more key permutations in between states, and adding an additional byte substitution procedure to the key schedule.

The efficiency of the application's encryption/decryption operation was measured using time and throughput. The avalanche effect and randomness tests were also used to assess the trustworthiness of the modified AES algorithm.

The paper has also argued that AES has different attractive advantages like high security, high throughput, and can be easily implemented both in hardware and software.

However, with the development of new computing concepts that may weaken the strength of current and standardized cryptography, the need to strengthen and provide modifications and diversities in cryptography are being widely accepted.

Some studies have proposed modifications to AES: the replacement of mixcolumn transformation to bit permutation to speed up the encryption of texts and images was proposed.

However, the modified AES algorithm with six round iteration is vulnerable to many security concerns such as brute force attacks, saturation assaults, square attacks, and Biclique attacks, necessitating the introduction of a stronger mechanism.

To summarize, the majority of the publications mentioned above focused on integrating two or more algorithms to improve security.

Some of the articles also intended to evaluate the same algorithm on various platforms and application platforms in order to enhance effectiveness without adding new value to the algorithms.

There is just one study relating to the issue of this thesis that focuses on reducing the number of rounds of the AES algorithm from ten to six, however as researcher [22] points out, lowering the round repetition to six rounds exposes the algorithm to tangible security assaults.

As a result, the goal of this thesis is to create a hybrid method that combines the modified AES with 8 round iteration with the MD5 hashing algorithm to address the gap and vulnerabilities in AES on key exchange between transmitter and receiver.

Chapter Three

3. methodology

This chapter specifies the research design used, data source, methods used for data analysis, data processing and analysis.

3.1 General approach

In this thesis, an experimental research design method was applied, the method is used to test the performance of the modified AES algorithm through measuring the execution time and through put and compare with the conventional AES algorithm. The modification will be done through minimizing the round key of the AES algorithm from 10 to 8. An interview also made with a business-to-business type of firm to discuss about how an online business transactions are being done using an e-commerce business sites and the problems they are facing in the area.

For AES, the 10 number of rounds in theory ensures a large enough security margin which was not affected for the past 20 years. On reducing the number of round iteration to 6 round can be affected by the best known attack the 'saturation attack or square attack short-cut attack. The newly modified AES-128 with 8 round iteration guarantee two additional rounds of security margin to ensure the security of the algorithm and also reduce the encryption and decryption time of the algorithm. [22]

About six files of different size and data type were selected randomly to be encrypted and decrypted by the conventional AES algorithm with 10 round and also compared with the modified AES algorithm with only six round and measure the performance of the modified AES algorithm. Finally the result of the experiment will be further demonstrated using quantitative analysis method through different graphs and charts.

3.2 Specific research design

To conduct this thesis primarily an experimental research method was applied due to the nature of the study to test the performance and strength of the modified AES algorithm through executing the algorithm at the encryption and decryption procedures using a client and server side in python programming language.

In addition, both descriptive and quantitative research types were also used to assess the performances of the AES and the MD5 cryptographic algorithms based on the result identified on the analysis and moreover quantitative research method used to

quantify the performance of the algorithm and to compare the result and summarize the total result of the algorithm and to explain the findings using various graphs and charts.

3.3 Data Collection Methods and Approaches

Thus, this thesis was mainly based on primary data that were obtained by random selection of messages to encrypt and decrypt in order to measure the performance and strength through different size of data, therefore, eight files of size of approximately two text files with 100KB and 1MB, two PDF files with 100KB and 1MB, two image files with 100KB and 1MB and two audio files with 100KB and 1MB were tested in the experimentation to measure the effect of file size and data types on the result of the modified algorithm and the implementation through creating a graphic user interface that allow us to upload each file at a time to encrypt and decrypt using the modified AES and the conventional AES algorithm to measure both execution time and through put using python.

3.4. Experimental Process and Tools

Experimental research methodology was employed to evaluate the strength of the newly modified AES algorithm of 128 bits size which cannot easily penetrated by a hacker and cracker using a python program which has its own feature tools and techniques.

The following step are used on testing the performance of the algorithms:

1. A python component packets of the GUI programming with TKinter and the java Crypto Utils packages were used to create an interactive interface which the user can upload the six files one at a time with different file size and data type, the user also expected to input the secret key so as to do the encryption and decryption operations.

2. The application then prepares the file and the key for encryption by converting them to their hexadecimal equivalent. The hexadecimal values are then fed to the Modified Reduced Advanced Encryption Standard algorithm (MRAA) to transform the file into an unreadable and secured format. Subsequently, the encrypted file is saved with the same file extension as with that of the original file for storage or transmission.

3. To recover the original file, the secret key will be encrypted using the MD5 algorithm and sent to the receiver with the encrypted data messages then the receiver selects the encrypted file and inputs the appropriate hashed secret key. The encrypted file and the key are then converted into their equivalent hexadecimal values and passed to the MRAA for decryption.

The above steps can be further demonstrated using the following standard diagram of the conventional AES diagram.

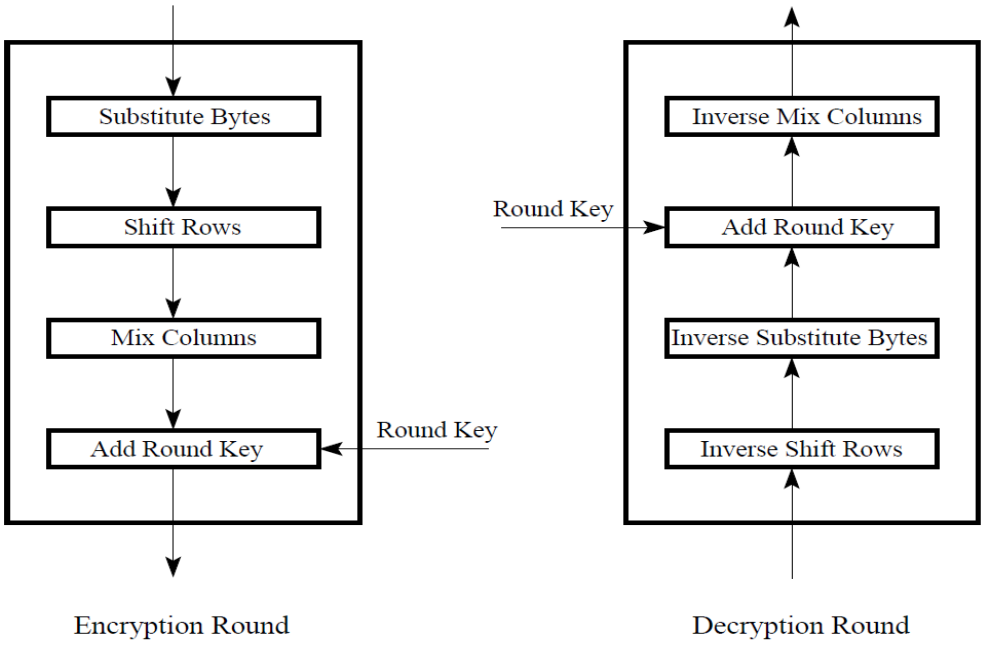

Figure 2. Shows the different steps that are carried out in each round

3.4.1 Python in cryptography

Using Python for cryptography is simpler than using languages such as C or C++; while free libraries such as OpenSSL are available, their use can be quite complex. Python removes these complexities with many built-in libraries that aid in cryptography scripting. It is also a great choice because Python is free in terms of license. Python can be described as an open-source, general-purpose language that is object oriented, functional, and procedural, and it allows for the interface with Python GUI Programming with Tkinter.

Up until now, the only way our programs have been able to interact with the user is through keyboard input via the **input** statement. But most real programs use windows,

buttons, scrollbars, and various other things. These widgets are part of what is called a Graphical User Interface or GUI.

3.4.2 Java Crypto Package

A java crypto package is one of a tool used to write, test and implement cryptographic algorithm in an AES encryption and decryption process.

In this thesis the Java crypto package is used to encrypt and decrypt some of the sample files of text type, PDF and audio and the encryption and decryption results are included in the thesis to measure the execution time of each of these files and used to compare their execution time.

One of the essential advantage of java over that of the python programming language in terms of encryption and decryption process is that Java will compute the time taken by the program to encrypt a given file and also to decrypt the time taken for both activities will be issued by the program. Since this thesis intended to reduce the execution time of AES algorithm by reducing the iteration round from 10 to 8, another operation or mechanism to measure execution time would not be required since java program itself provide the execution time.

3.4.3 Message Digest 5 (MD5)

MD5 is a one-way hash function that takes an input and generates a hash value. It gives you a technique to encrypt any messages you send and a way to verify the validity of any data you send between servers. Despite the fact that MD5 has been discovered to collide, it remains one of the most extensively used hash functions in the world. One of the most significant uses of MD5 is to ensure that data is safely moved between servers. We can check the integrity of files transported between servers by generating MD5 hashes on both sides and comparing them. The data transmission was successful if the hash value matched.

MD5 is a one way hashing function meaning that one can create a hash value from a message but cannot recreate the message from the hash value.MD5 creates a 128-bit message digest from the data input which is typically expressed in 32 digit hexadecimal number.[18]

3.5. System Architecture

The encryption process was designed to secure user's file supposed as confidential; the confidential files can be of any type and format. The system was developed using python programming language.

The user will select a confidential file of different data type and file size as two text files with 100KB and 1MB, two PDF files with 100KB and 1MB, two image files with 100KB and 1MB and two audio files with 100KB and 1MB were tested in the experimentation to measure the effect of file size and data types on the result of the modified algorithm and input the secret key for the file, and the application then prepares the file and the key for encryption by converting them to their hexadecimal corresponding. The hexadecimal values are then fed to the MRRA algorithm to transform the file into cipher text and secured format. Then encrypted file is saved with the same file extension as with that of the original file for storage or transmission.

To recover the original file, the user selects the encrypted file and inputs the appropriate secret key. The encrypted file and the key are then converted into their equivalent hexadecimal values and passed to the MRRA for decryption.

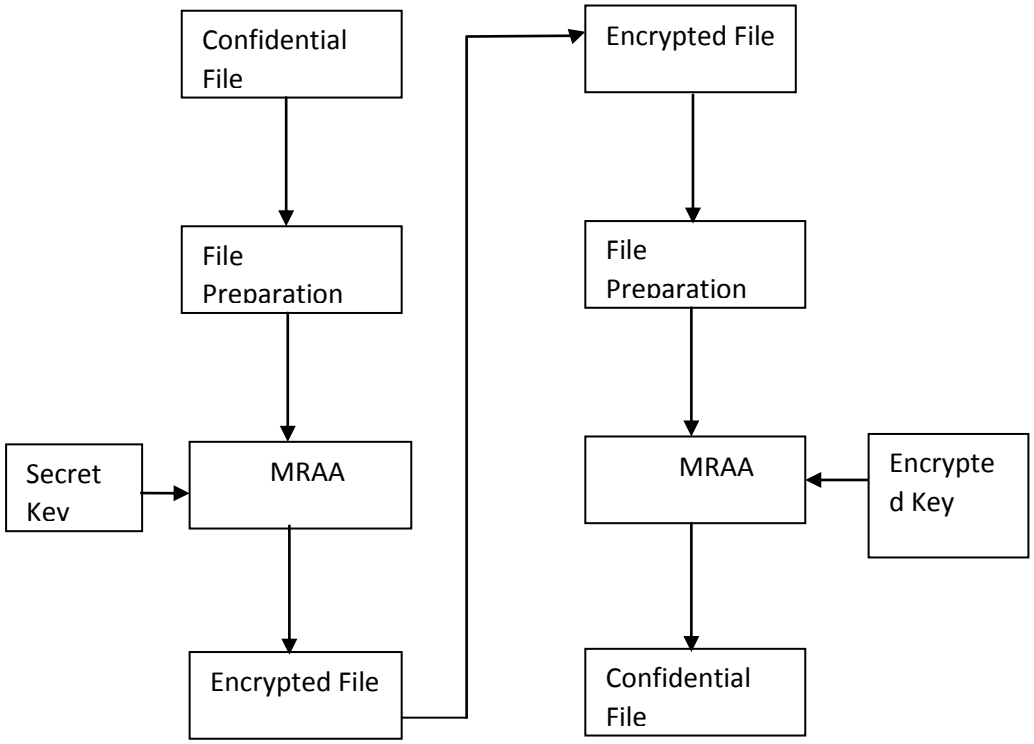

Figure 3: Block diagram for Encryption and Decryption process.

3.6 AES Algorithm Description

Advanced Encryption Standard (AES) refers for Advanced Encryption Standard. Rijndael is another name for it. NIST (National Institute of Standards and Technology) created it with the goal of securing confidential but unclassified information held by US public agencies. It is presently utilized all around the world.

To encrypt data with AES requires only one private key (symmetric key). The approach uses a single standard irreducible polynomial of degree '8' to compute multiplicative inverse tables.

Each layer in the algorithm involves with S-boxes and inverse S-boxes to function. In our implementation, instead of one irreducible polynomial of degree '8,' we employ sixteen irreducible polynomials of degree '8'. AES keys are commonly 128, 192, or 256 bits in length; we utilize a 128-bit key.

It is founded on the substitution-permutation network principle. It has a 128-bit constant block size and 128, 192, or 256-bit variable key size.

3.6.1 Message Digest 5 (MD5) algorithm description

MD5 (Message-Digest Algorithm 5) is a frequently used cryptographic hash function with a 128-bit message digest and is an Internet protocol. This has been used in a number of different security applications. The primary MD5 algorithm uses a 128-bit word that is partitioned into four 32-bit terms.

This thesis will further demonstrates the combinations of the best features of both symmetric and asymmetric encryption techniques, hence the data (plain text) that is to be transmitted is encrypted using the AES algorithm. The data (plain text) used input to MD5 to generate AES key. This key encrypted by using modification of Diffie Hellman (MDF). MD5 algorithm is used to ensure integrity of the data that is transmitted through the e-commerce sites, in addition it also easily generate secret key used in AES algorithm.

As a result, the client provides the message's cipher text, as well as the AES key's cipher text and the message digest's cipher text. The signal, as well as the cipher text of the AES key, is decrypted using (MDH) at the receiver side to retrieve the AES key. This could then be used to decipher the message's encrypted text.

AES decryption to obtain the plain text. The plaintext is again subjected to MD5 hash algorithm to compare with decrypted message digest to ensure integrity of data.

3.7. Validation and Evaluation Process

The algorithm used to optimize the functionality of the AES encryption algorithm is used to evaluate the correctness of this research topic. AEs is a standard algorithm being allowed to serve to protect the security of useful data items being stored within the database and also on transmitting data items through the global network without losing its integrity and confidentiality.

This thesis took this standard algorithm and intended to modify by reducing the number of iteration from 10 to 8 without compromising its security and reliability, in addition the newly modified algorithm were combined with another algorithm the MD5 which avoids the limitation of AES algorithm on exchanging key between the sender and receiving party.

The goal and objective defined at the initial stage of this thesis were met by using an appropriate methodology and hence the result of this study could be operational.

The evaluation process were done on selected sample files running on the standard algorithm and the execution time were measured through calculating the elapsed time by deducting the starting time from the end time and compared with the standard AES algorithm.

Chapter Four

4. Experimentation and Result

4.1. Model Description

Multimedia data items like text, images, audio, animation and video are being frequently transmitted across the electronic commerce transactions throughout the global online market day-to-day. Therefore, it is very essential to protect valuable business transactions including monetary transfer against unauthorized users and harmful attacks.

Thus the intention of this thesis is to ensure the security of this multimedia data items and enhance the execution time of the algorithms on encrypting and decrypting of a message which are transacted on e-commerce using cryptographic techniques.

The two essential activities which are done in cryptography are to encrypt a message or a file which entails converting a plain text data into a cipher data using a cryptography algorithm and encryption key. The other activity is to decrypt, which involves converting the cipher data into a plain text data using the same algorithm and a decryption key.

This thesis is to implement a modified Advanced Encryption Standard (AES) algorithm through reducing number of round key of the AES from 10 to 8 which will enhance the performance of the algorithm by reducing execution time and ensuring the security of the multimedia data (such as text, image and video) which will transact in the global market through e-commerce. [21]

This thesis was focused on the following primary four steps undergone in symmetric key cryptography (AES) which will be implemented in each round of AES:

(1) Substitution,

(2) Shift rows,

(3) Mix columns, and

(4) Add round key.

The above four steps are modified in the way implemented in this new AES algorithm from 10 round iteration to 8 rounds and from 43 words in the conventional AES word size to only 35 words which were reduced as the reduction of the number of rounds of the MRRA.

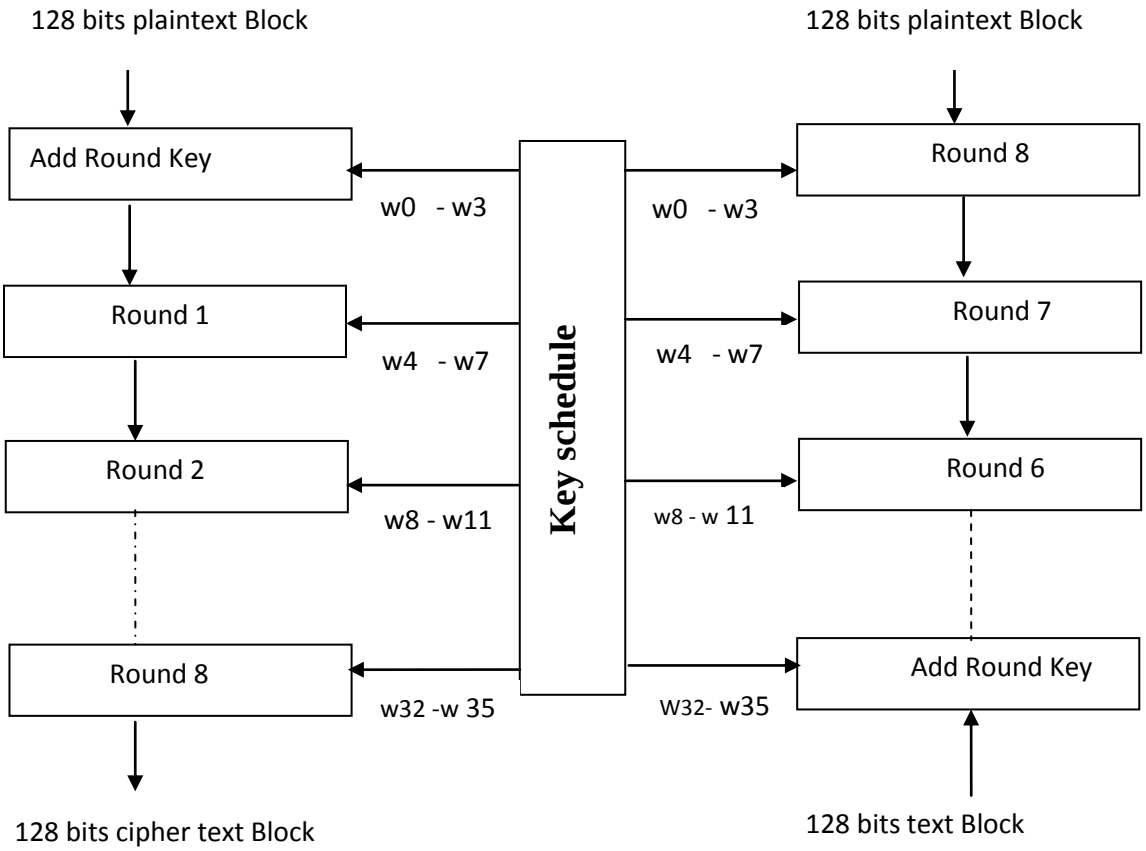

Figure 4. The overall structure of AES encryption/decryption

Therefore, when we reduce the number of rounds from 10 round to eight round all the above steps could be done only with eight rounds instead of ten, which significantly reduce the execution time of the algorithm and through put on sending and receiving of a message in a business transactions.

The following flow chart diagram will further demonstrate the processes of the AES algorithm:

4.2 Modified AES Algorithm Flow chart

The flow chart below gives an overview of the algorithm.

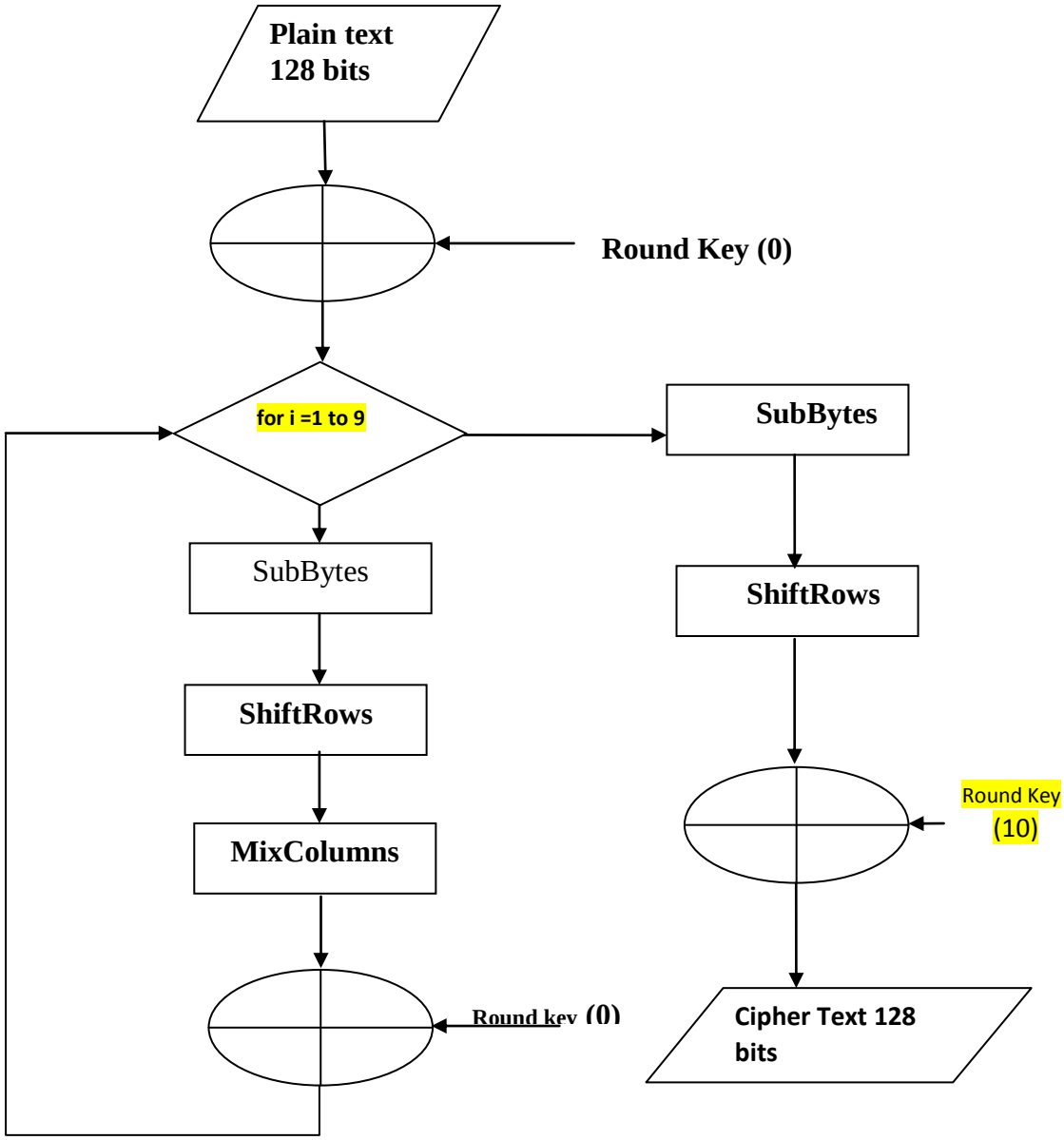

Figure 5. Modified AES Flow chart

4.3 The Four Steps in Each Round of Processing

STEP 1: SubBytes for byte-by-byte substitution during the forward process.

- The corresponding substitution step used during decryption is called InvSubBytes.
- The python implementation uses a four loop to construct a 256 element array of lookup values for integers ranging from 0 to 255. For every integer within the range of 0 to 255, we first find its multiplicative inverse in GF (2^8), then we XOR the result using four different circularly rotated versions of the result, then XOR the result using the constant C.

The same thing will be done for the decryption lookup array, except that we first do the XORing and compute the multiplicative inverse.

STEP 2: is called ShiftRows for shifting the rows of the state array during the forward process.

- The corresponding transformation during decryption is denoted InvShiftRows for Inverse Shift-Row Transformation.
- The Shift Rows transformation has the following important steps:

➢ Not shift the first row of the state array at all.
➢ Circularly shifting the second roe by one byte to the left.
➢ Circularly shifting the third row by two bytes to the left.
➢ Circularly shifting the last row by three bytes to the left.

Knowing that the input block is written column-by-column, which means that the first four bytes of the input block contain the first column of the state array, the next four bytes contain the next column, and so on. As a result, rearranging the rows in the manner specified jumbled up the input block's byte sequence.

- The shifting steps in the decryption process shift the rows in the exact opposite way. The first row is unchanged, the second row is shifted to the right by one byte, the third row is shifted to the right by two bytes, and the last row is moved to the right by two bytes, and the last row moves to the right by three bytes.The goal of this transformation is to scramble the byte order inside each 128 bit block.

STEP 3: is called MixColumns for mixing up of the bytes in each column separately during the forward process.

- The corresponding transformation during decryption is denoted InvMixColumns and stands for inverse mix column transformation.
- The purpose is to further jumble the 128-bit input block, and this step does so by replacing each byte of a column with a function of all the bytes in that column.This is to mean that each byte in a column is replaced by two times that byte, plus three times the next byte, plus the byte that comes next, plus the byte that follows within the same column circular.
- The shift rows step along with the mix column step causes each bit of the cipher text to depend on every bit of the plaintext after 8 rounds of processing.

STEP 4: The fourth step is AddRoundKey for adding the round key to the output of the previous step during the forward process.

- The corresponding step during decryption is denoted InvAddRound-Key for inverse add round key transformation.
- The round key for each iteration is obtained from the original 128-bit encryption key. The XORing of the round key with the state array is one of the four steps of each round for encryption and decryption.From the original 128 bit encryption key, the AES key Expansion method is utilized to generate the 128 bit round key for each round. The logic of the key expansion technique is meant to ensure that if one bit of the encryption key is changed, the round keys will be affected for several rounds.in the same way that the 128 bits input block is arranged in the form of a state array, the algorithm first arranges the 16 bytes of the encryption key in the form of a 4*4 array of bytes as shown below:

$$\begin{pmatrix} K0 & K4 & K8 & K12 \\ K1 & K5 & K9 & K13 \\ K2 & K6 & K10 & K14 \\ K3 & K7 & K11 & K15 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} W0 & W1 & W2 & W3 \end{bmatrix}$$

Therefore, the first four bytes of the encryption key constitutes the word W0, the next four bytes represent the word W1, the third four bytes represent the word W2 and the last four bytes represent the last word W3.

The algorithm therefore, expands the word [W0, W1, W2, W3] into a 44 word Key Schedule that can be arranged as:

[W0, W1, W2, W3 ... W43]

Before the round-based processing begins, the words W0, W1, W2, and W3 are bitwise XOR'ed with the input block. In each of the eight rounds, the remaining 40 words of the key schedule are used four at a time. The preceding two facts are likewise true for the decryption procedure, with the exception that the key schedule is now reversed in order.

Before any round-based processing, the last four words of the key schedule are bitwise XOR'ed with the 128-bit cipher text block. Following that, in each of the eight rounds of processing, each of the four words from the remaining 40 words of the key schedule is employed.

AES has no weak keys thanks to the key expansion mechanism. A weak key is one that decreases a cipher's security in a predictable manner.

4.4. Modified Reduced-Round AES Algorithm (MRRA)

The proposed file encryption technique was based on a modified AES. There were three changes to AES: (a) more add round key operations between the stages of the conventional AES cipher round; (b) more byte substitution operations and round constant addition in the key schedule algorithm before the key expansion process; and (c) fewer round iterations from 10 to 8.

Two activities were added to the AES cipher round: AddRoundKey utilizing the XOR function was added after the SubBytes process in the first five rounds, and ModAddRoundKey based on modulo addition was added after the ShiftRows process in the last five rounds. After the SubBytes procedure in the previous round, another ModAddRoundKey is added. SubBytes to InvSubBytes, ShiftRows to InvShiftRows, MixColumns to InvMixColumns, and ModAddRoundKey to ModSubRoundKey will be the inverse of encryption processes.

The diagram below is designed to demonstrate the newly modified Reduced Round AES algorithm only with eight rounds and the implementation of the algorithm at every step of the entire operation.

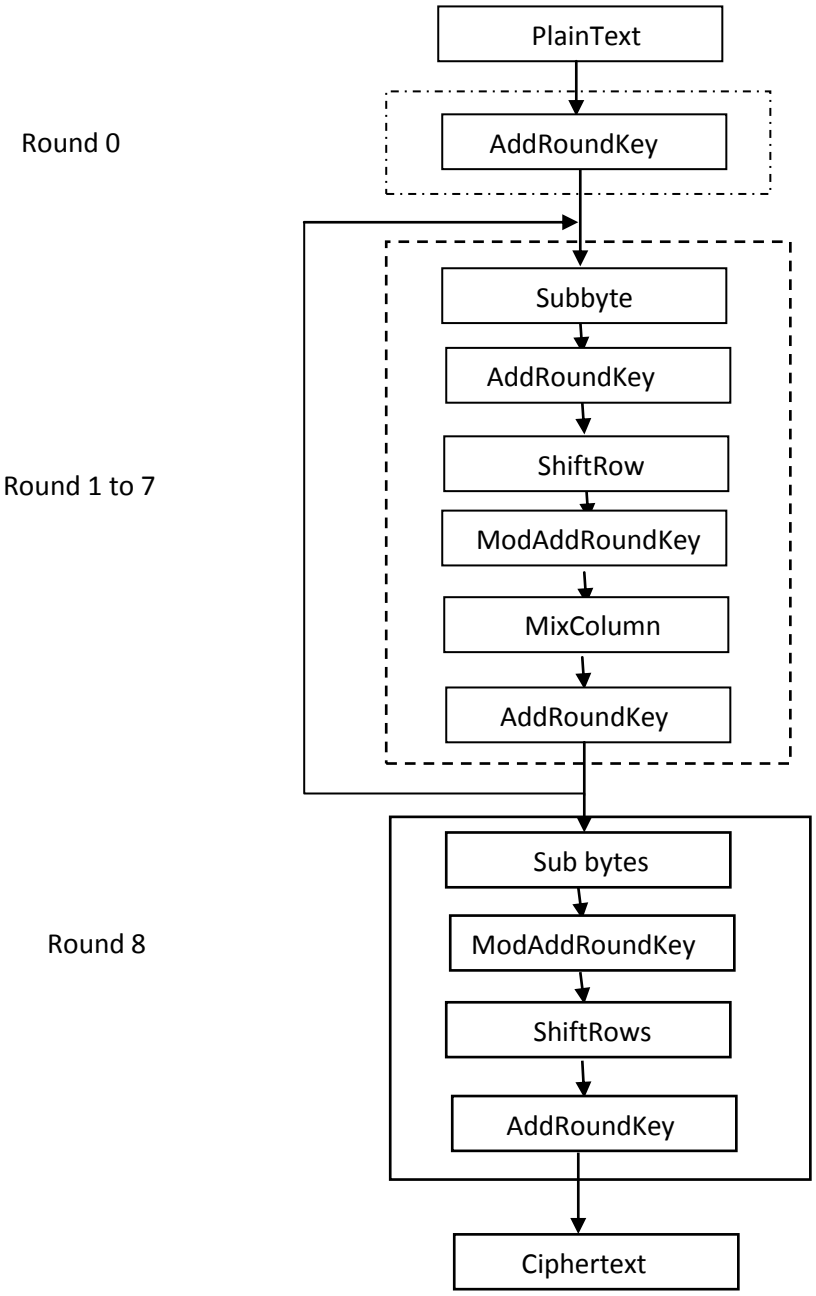

Figure 6. MRRA Cipher Round Algorithm

4.5. Experimental analysis

For implementing the above model six files of different size and different data types were taken and tested using python program based on the following step:

1. A file with 10kb of textual data type were loaded to the program and executed using the standard AES algorithm and the encrypted time were measured and stored.

2. The same file size data of image format were loaded to the program and encrypted using the secrete key and the execution time were taken.
3. The same file size data item with audio type were loaded to the AES standard algorithm and encrypted with the secret key and the execution time were measured and stored.
4. The same file size with a PDF data type were loaded to the algorithm and executed with the secret key and the execution time were measured and stored.

The above step was repeated about eight times with files of two text files with 100KB and 1MB, two PDF files with 100KB and 1MB, two image files with 100KB and 1MB and two audio files with 100KB and 1MB are taken for the experiment to compute their execution time were collected and stored at every step of the operation.

Table 1: Sample Test files

File	**Type**	**Size**
1	Text	100KB
2	Text	1MB
3	PDF	100KB
4	PDF	1MB
5	Image	100KB
6	Image	1MB
7	Audio	100KB
8	Audio	1MB

The table below shows different data types with different time executed with the standard AES algorithm and measure the execution time in millisecond.

Table 2. Execution time for AES encryption standard algorithm in millisecond.

Execution Time of the Standard AES Algorithm				
File Size	**Data Type**			
	Text	**Audio**	**Image**	**PDF**
100 KB	4803.4	5721.6	5127.3	4830
1MB	47189.2	58589.2	58511.4	49459.2

The graph below shows the AES standard algorithm execution time for various data types which are taken as a sample file which are selected for this thesis purpose and an experiment were conducted through the same size (100kb) of different data types (Text, Audio, Image, and PDF) executed in a python program. The result shows that textual data items have less execution time than PDF files, PDF files have less execution time than image files and also image files have lesser execution time than Audio type of file.

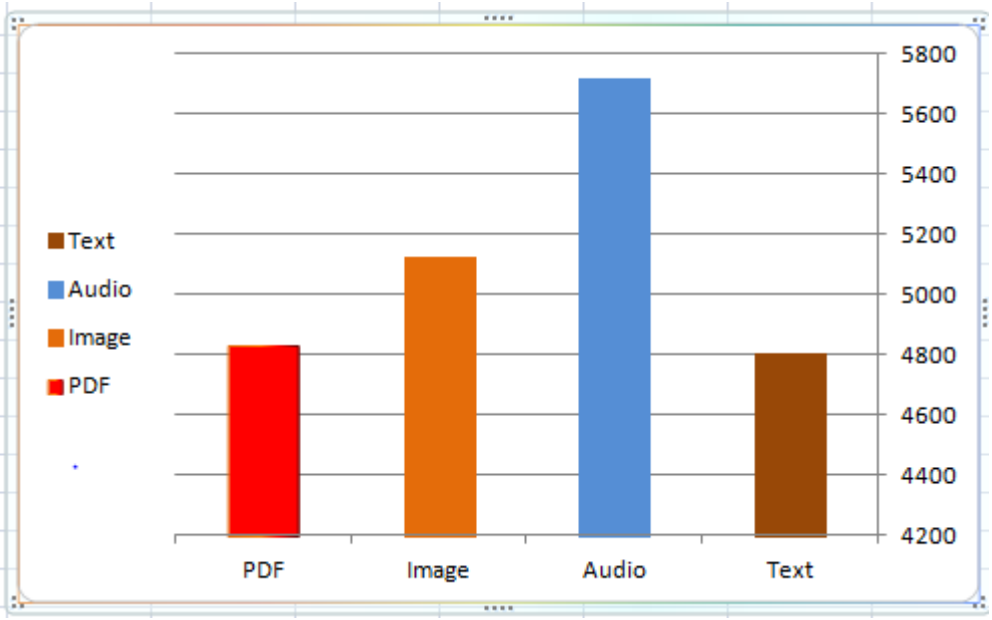

Figure 7. AES encryption standard algorithm execution time of 100KB in millisecond.

The table below also shows that as we increase the file size from 100KB to 1MB of the sample files with different data type as indicated on the above example still Audio data execution time goes higher than the Image file, the image file execution time is higher than PDF file and also the PDF file with the same file size executed with text file the PDF file execution time is higher than text file.

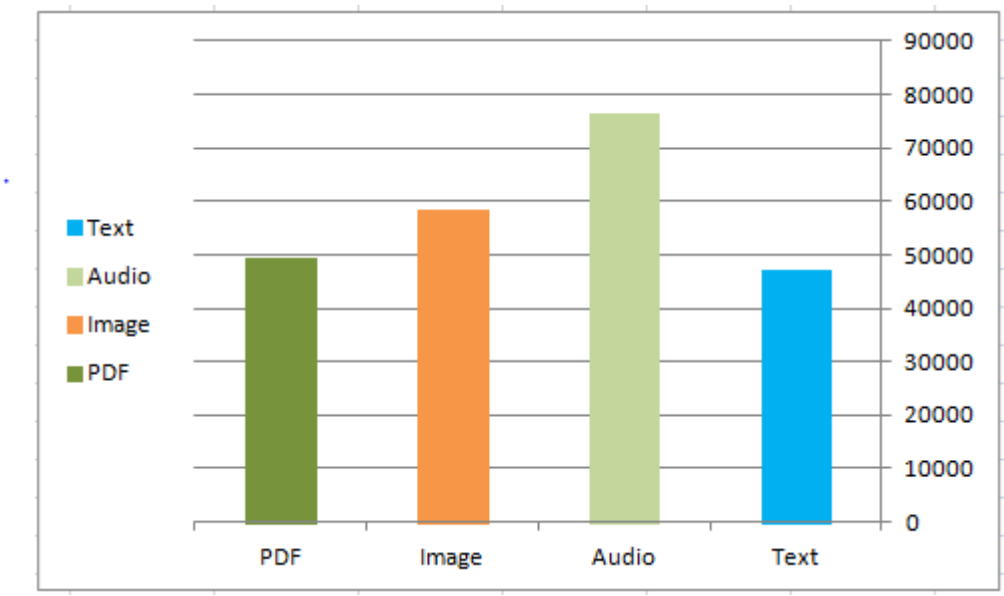

Figure 8. Standard AES encryption algorithm execution time of size 1MB in millisecond.

In the experiments, the secret keys used were "EH35@r3mcD1ccMW", Different file types and sizes were also used in the experimentation as shown in the above Table.

The hardware component used for the experimentation is an Intel(R) Core™ i3-2330M and CPU with 2.20 GHz and 6GB of Installed Memory (RAM). Each sample test file is encrypted with the cipher keys mentioned above and the result were used to measure the execution time of the MRRA through encrypting and decrypting the files with different data types.

On measuring the performance of the newly modified AES algorithm on encrypting and decrypting files using the cryptographic transformation, each of the above files with different data types is encrypted six times The average result of the experimentation of the six files were taken as the encryption and decryption time for each sample file. The average time was taken to measure the performance of the modified algorithm.

4.6. Password hashing using MD5 algorithm

The primary focus of this paper is to enhance the performance of AES algorithm using a reduced-round approach and also to maintain the security of AES through combining it with the MD5 hashing algorithm.

Since AES algorithm has a limitation on securing the key upon exchanging it between the sender and the receiver through the various networking communication media.[11] Hence this thesis focuses on avoiding the above limitation by enabling the key exchanging mechanism more secured by using the hashing algorithm of SHA256 as demonstrated below using a python code.

The key selected for the AES encryption and decryption algorithm for this experiment is:-

'yishaklakewfromst.marryuniversitycomputersciencedepartment'

The hashing algorithm produced an encrypted equivalent of the above password as indicated as an output of the python code below.

'tMWvaBQCYzrXcpjb1A6gjQyrp9CU0_D3vTi8Ir22WXQ='

```
password_provided = "password"
password = password_provided.encode()
salt = b'yishaklakewfromst.marryuniversitycomutersciencedepartment'
kdf = PBKDF2HMAC(
     algorithm = hashes.SHA256(),
     length=32,
     salt=salt,
     iterations = 100000,
     backend=default_backend()
)
key = base64.urlsafe_b64encode(kdf.derive(password))
print (key)
```

```
b'tMWvaBQCYzrXcpjb1A6gjQyrp9CU0_D3vTi8Ir22WXQ='
```

Figure 9. Hashing encryption key using SHA256.

4.7. Result and analysis

The sample files selected for the encryption and decryption process by this research thesis are taken and encrypted using the Java cryptoutil package and the result of the encryption process on some sample file are displayed in the figures below.

The following output figure indicates the java crypto package written and executed on Java program netbeans to show a simple example of the encryption algorithm result when sample file of PDF file is encrypted.

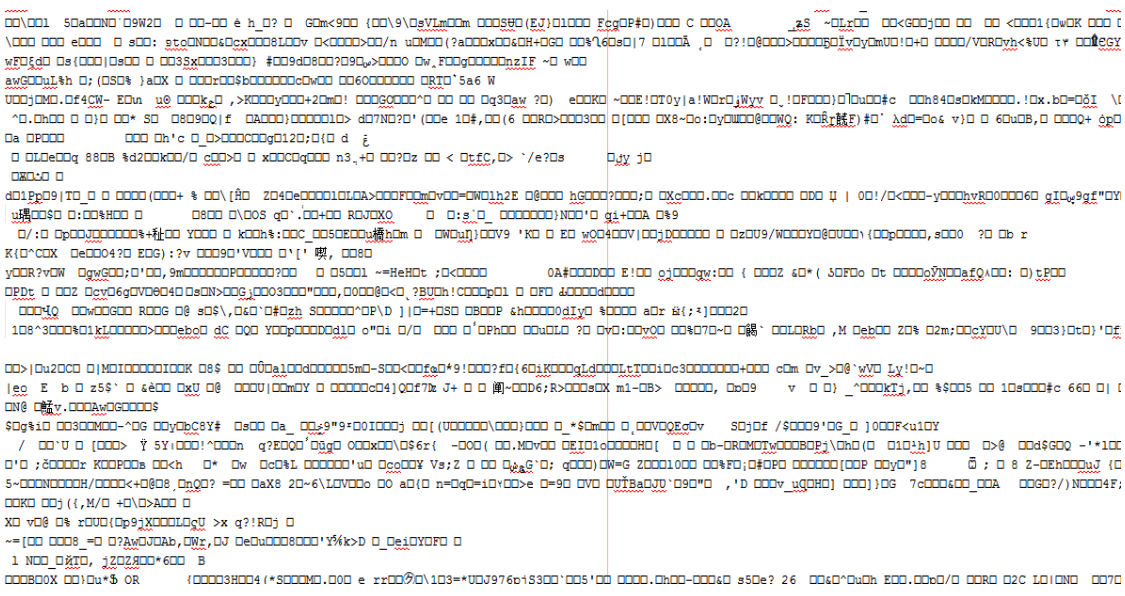

Figure 10. Encryption result of PDF file

The figure below also shows the result of the java cryptoutil which encrypt and decrypt a file given by the user. Here a sample audio file of size 1MB is given to the program and the program has executed the file and the cypher file is displayed below.

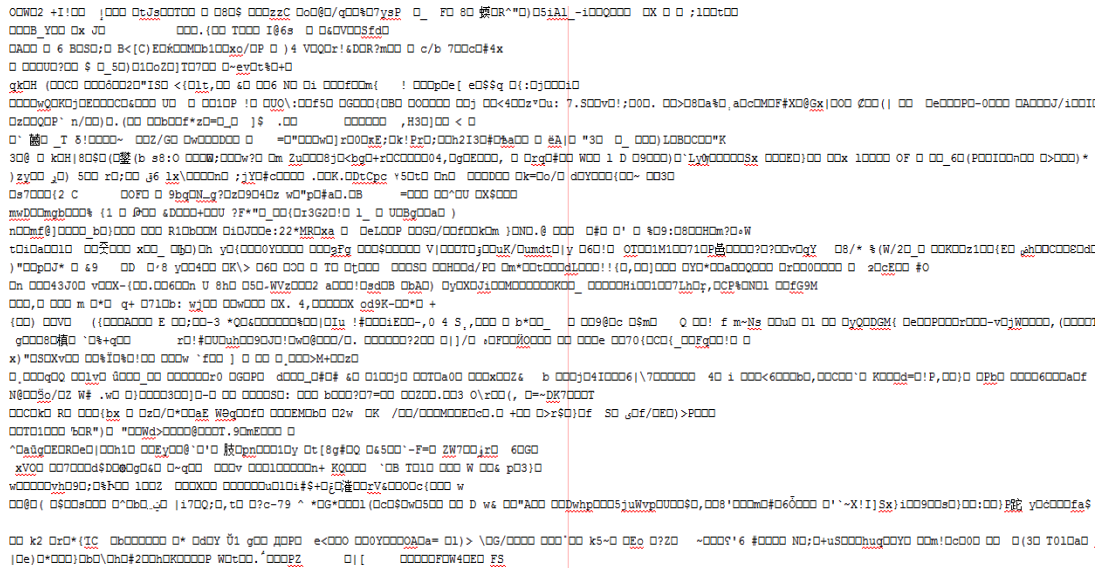

Figure 11. Encryption of sample Audio file.

The figure below demonstrate the encryption process result of a text file with size of 1MB from the sample files used in this experiment. The same task were made for all sample size of 100MB and 1MB with different type of file of Text, PDF and Audio and the results were taken and execution time were collected and measured.

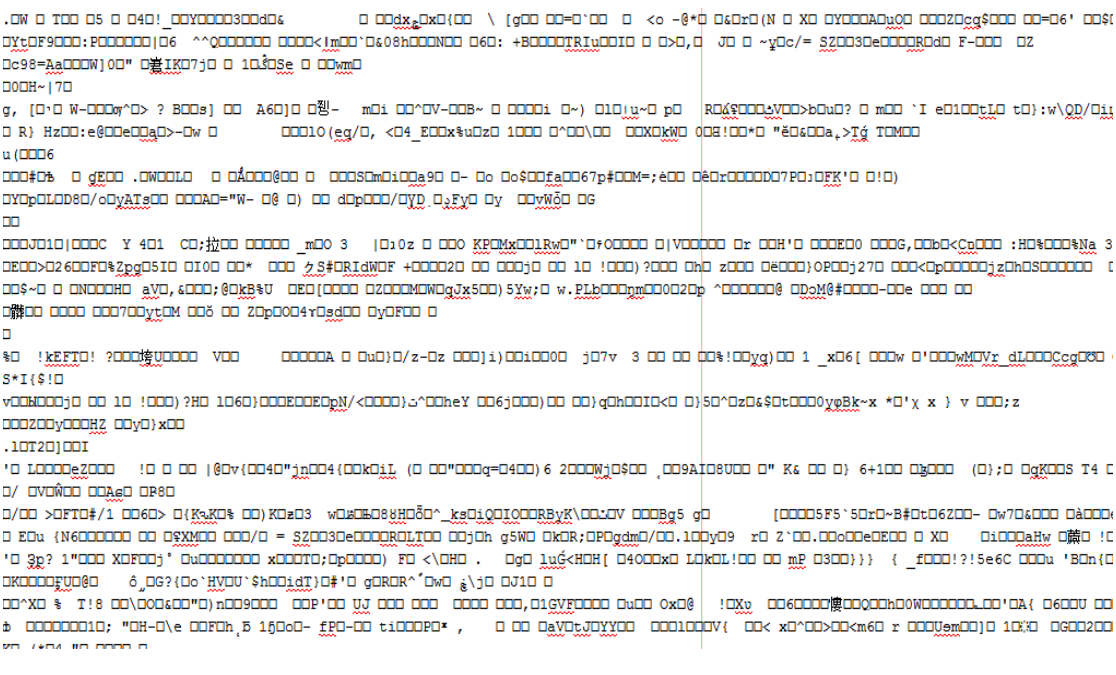

Figure 12. Encryption of sample Text file.

The above hexadecimal representations are 89 round keys, each of them are 128 bit long, for the 8 rounds that will be used when the user supplied encryption key is 128 bits long. By the time, the first round key listed above is what you need to XOR the input block with before any encryption rounds. Eight files were encrypted using the two cipher keys mentioned above using the standard AES algorithm, and the MRRA algorithm, portions of the cipher texts for the first three files.

Metrics

The following defines the different tests that were used in the evaluation of the MRRA.

The performance of the file cryptographic transformation is analyzed using the metrics time and throughput. The evaluation parameters are as follows:

a) Encryption Time – The required time to process the transformation from plaintext to its equivalent cipher text and measured in milliseconds.
b) Decryption Time – The time to recover back the original plaintext from the cipher text and measured in milliseconds.

4.7.1. MRRA AES encryption experiment

Eight files were encrypted using the cipher key mentioned above using the standard AES algorithm, and the MRRA algorithm, portions of the cipher texts for the first three files were presented in the following table.

Table 3: sample cipher

File	Key	
	AES	MRRA
1	A23d345fd11fg45g56d32wjh D00321dsa6700mb1100wq32 Wq2310097rre56220	Mn89090923wasf2d32zx4 K6h899gh000kloas2312q
2	Uu565g01209dj45kn55112ww2 Er22tg4hj8k009lok07uih0902wa Asd3243ugc342gbf0980knm0	45yht431kbnv098080hb0 Rr54thgj706050fd99hgtj5 Df42ws
3	Rt54ersd328iu09080j0k0lbnh67 Nmn55g332fdgs34298hk776yg 78yh9nbg555776rfde89800980 We12sd43d7	Qm231as564dcsg45fgh89 jnh09d34weq22655jhu67 nddsaw34fg88
4	Lk44ou?sda2332jkh??ok033we sa0lk554rfg6rt3434dcsc786kk8 wesa324asd32fd767niihh76gbn 8m89cvd5879gh00kjl00rws232	ss79hgy??kl87lk66bb43 hgn754??jkhgm65cc5y cf341d67gh500gh90u 7?k

The table bellow shows that the Encryption and decryption processes which was done for the Modified Reduced Round AES (MRRA).

Table 4. MRRA encryption execution time.

Execution Time of the Modified AES (MRRA) Algorithm				
File Size	Data Type			
	Text	**Audio**	**Image**	**PDF**
100KB	4418.7	5417.5	4852.1	4548.1
1MB	45247.5	55475.2	49685.5	46572.7

The performance of the cryptographic application was also evaluated utilizing speed and throughput. The time taken to encrypt and decrypt a file was used as a metric for speed. The formula to compute for encryption or decryption time is as follows:

Elapsed time = End time – Start time

The total time consumed using the standard AES algorithm with the above four data types in 100KB file size registered about an average time of 50.95% (5095 millisecond) and with that of the MRRA with the same files in the same data size of 100KB took an average execution time of about 48.09% (4809 Millisecond) in millisecond. Therefore, the modified AES algorithm has improved the encryption time of those different files by 2.86%.

In another measurement by taking the above four different files of data types of text, image, audio and PDF with testing the execution time of the files with the same file size capacity of 1MB from the standard AES algorithm execution time which was 23174.9 millisecond were reduced by the modified AES algorithm to 19236.4, therefore, the MRRA has minimized the execution time by 3938.5 millisecond which constituted an improvement of the execution time by 9.85%. From these experimentation it is possible to conclude that as the size file to be encrypted or decrypted increases, the algorithm execution time also increases.

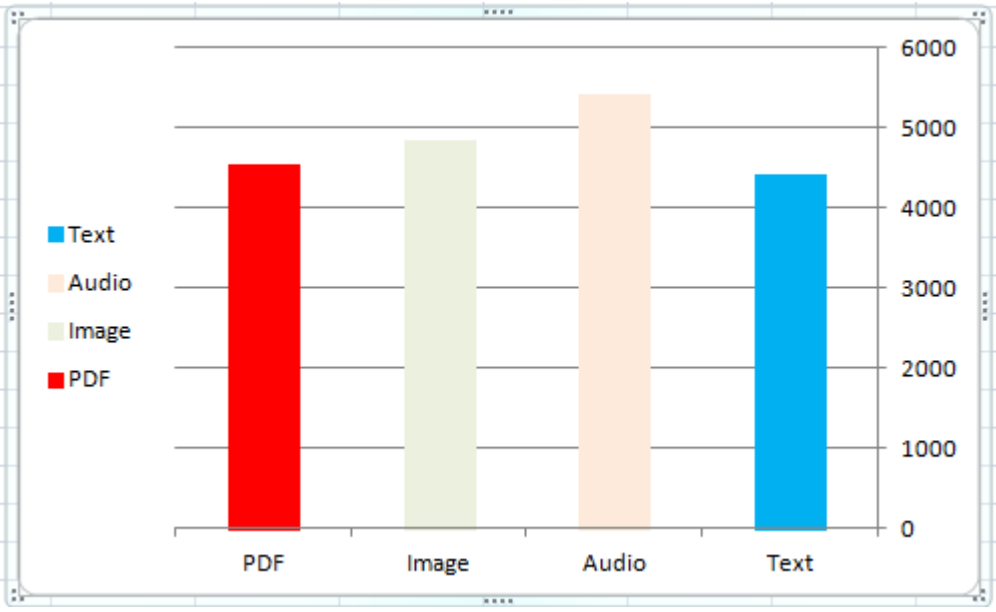

Figure 13. MRRA encryption File size compared with execution time.

As the figure above shows the same file size (100KB) of different data types of Text, Image, Audio and PDF files are taken to the experiment and the MRRA executed those files using the java programming language.

The execution time we got are somehow closer in those different data types, but when the file size of those sample files increases the execution time of the Audio data went up in a huge amount than the Image file, the image file execution time increase in large number than the PDF file and the PDF file execution time also increases at large than the Text file of equal size as indicated in figure 11.

The chart below indicates the modified AES algorithm execution time in a 1MB file size equally is given for the sample Text, PDF, Audio and Image files and the algorithm in java programming language were executed and the result were summarized using a chart below.

When we compare the result with the above table of 100KB file size, as the file size increases of the sample files, the execution time still is higher for Audio files then the Image files, the execution time of the image file with the same size with that of the PDF file and the PDF file execution time is also higher than the Text file of equal file size.

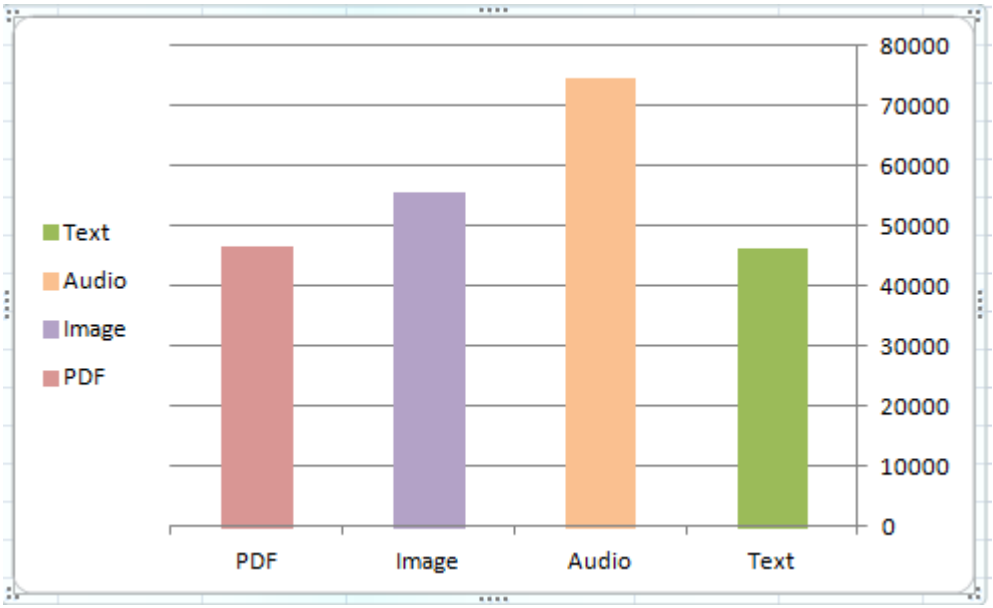

Figure 14. MRRA encryption execution time in millisecond

The relationship between the standard AES and the newly designed MRRA AES algorithm in terms of file size of 100KB with a file format of Text, PDF, Image, and Audio files and a result in a graph below were indicated.

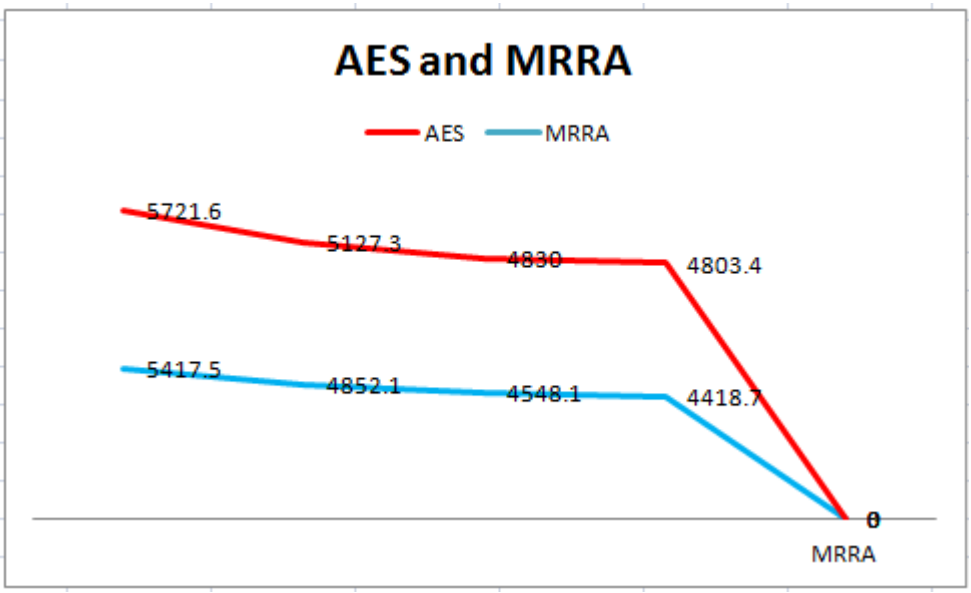

Figure 15. Relationship between MRRA and Standard algorithm of 100KB in millisecond.

The figure shows that the Standard AES algorithm has registered higher execution time with 10 round of iteration and the MRRA which the thesis focus has registered lesser execution time with sample files of the same file size of 100KB.

4.7.2. MRRA AES decryption algorithm experiment

The following table contains the standard AES algorithm decryption time of those sample files for file size of 100KB and 1MB file size of text file, PDF file, Image file and Audio files.

Table 5. Standard AES decryption algorithm execution time.

Execution Time of the Standard AES decryption Algorithm				
File Size	**Data Type**			
	Text	**Audio**	**Image**	**PDF**
100 KB	4083.1	4144.3	4123.7	4114.2
1MB	41810.9	42437.6	42226.7	42129.4

As the result above indicates the decryption process took less execution time than the encryption process of the standard AES algorithm in those different data types indicated in the sample file taken to the experiment with a file size of 100KB files of text, PDF, image and audio type. The same files are also taken and tested with a different file size of 1MB and the result is displayed.

The table above also indicates that text type files have less execution time than the others, PDF file format with the same file size with the other files has registered the second less time and then is image file formats the third lesser execution time and the higher execution time were registered by audio type of file format.

The result above indicates that as the file size increases the decryption execution time of those sample files will increase, as a reference when the file size of text increases from 100KB to 1MB the execution time increases by 37727.8 millisecond, the audio file by 38293.3 millisecond, image file by 38103 millisecond and PDF file increases by 38015.2.

The graph below shows the distribution of the execution time in relation to the sample files of text, PDF, image and audio in 100KB file size using a 10 round execution iteration of the standard AES algorithm.

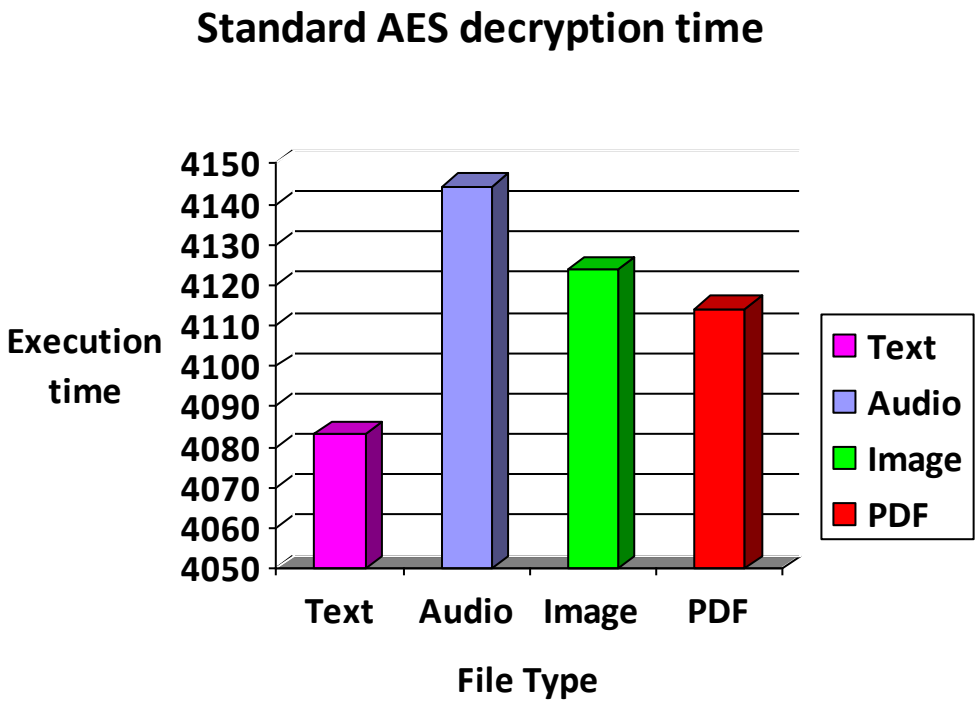

Figure 16. The standard AES decryption algorithm execution time in millisecond.

The other experiment done in this thesis was to decrypt the sample files encrypted above by first taking text file with 100Kb, PDF with 100KB, Image with 100Kb and Audio file with 100KB. Each are decrypted at a time with the encryption key which is hashed above by the MD5 hashing algorithm to ensure the key exchange between the sender and the receiver with an 8 round iteration of the MRRA.

To measure the execution time of all the files below which referenced in a table the start time of each algorithm were recorded in millisecond and also the end time of each file first with 100KB sample were also measured in millisecond and recorded, then to identify the elapsed time, the start time were deducted from the end time and the result displayed in a table for each file type were recorded in the table below.

The following formula was used to measure the execution time.

Elapsed time = End time – Start time

The following table demonstrate the execution time which was found by the MRRA with 8 round iteration and hashed by MD5 algorithm for each file format first with 100KB and the same operation was done with the same files in a different file size of 1MB. Then the elapsed time were computed with the formula above and the result were recorded in the table.

Table 6. MRRA decryption execution time in millisecond.

Execution Time of the MRRA decryption Algorithm				
File Size	**Data Type**			
	Text	**Audio**	**Image**	**PDF**
100KB	4018.2	4117.5	4082.1	4051.3
1MB	41146.4	42163.2	41800.7	41485.3

The table above demonstrate the output of the experiment done above and the result shows that the least execution time was registered by text file for both file sizes of 100KB and 1MB.

The next least time were registered by a PDF file with the same size, the next least execution time was registered by an image file with the same 100KB and the highest execution time was registered by the Audio file of the same file size.

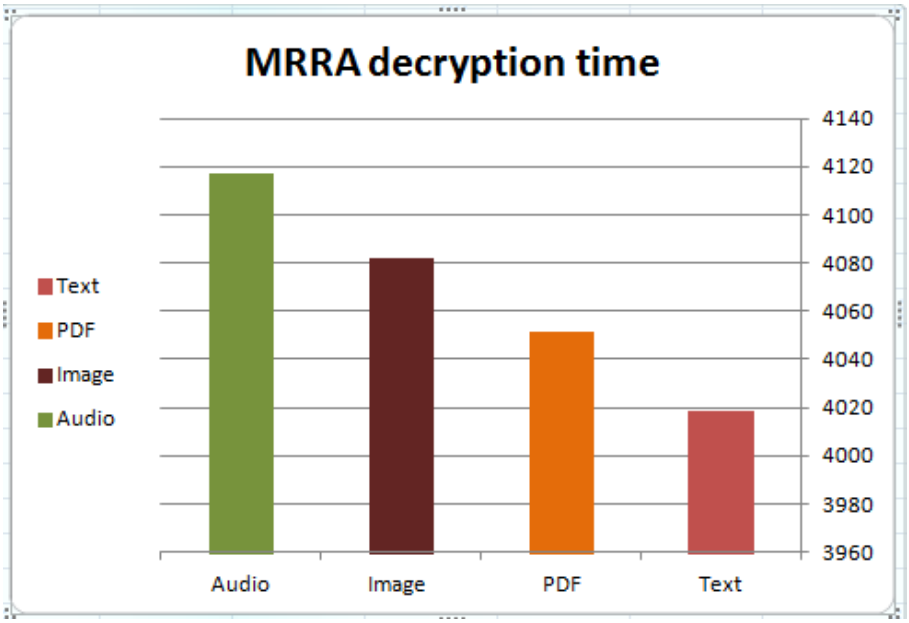

Figure 17. MRRA file decryption in millisecond.

The graph above demonstrate that the decryption time of the sample file with 100KB in 8 round of iteration experiment done with a Text, PDF, image and Audio file the result demonstrate that decryption process will consume less time than that of the encryption process with all the sample files.

The other experiment done in this thesis includes the decryption process of the sample files and evaluated with 1MB file size of each file using the Modified Reduced Round Advanced Encryption algorithm with an eight round iteration using a python program.

The process will take an input of first text file with size of 1MB and the stored hashed key which was encrypted using the MD5 hashing algorithm then a PDF file with 1MB then next an image file and finally an audio file formats were given to the algorithm and the execution time were computed and displayed in the table above.

The table values are displayed on the figure below that shows the execution time of the sample files with equal file size of 1MB and it demonstrates the change in file type has resulted with a different execution time.

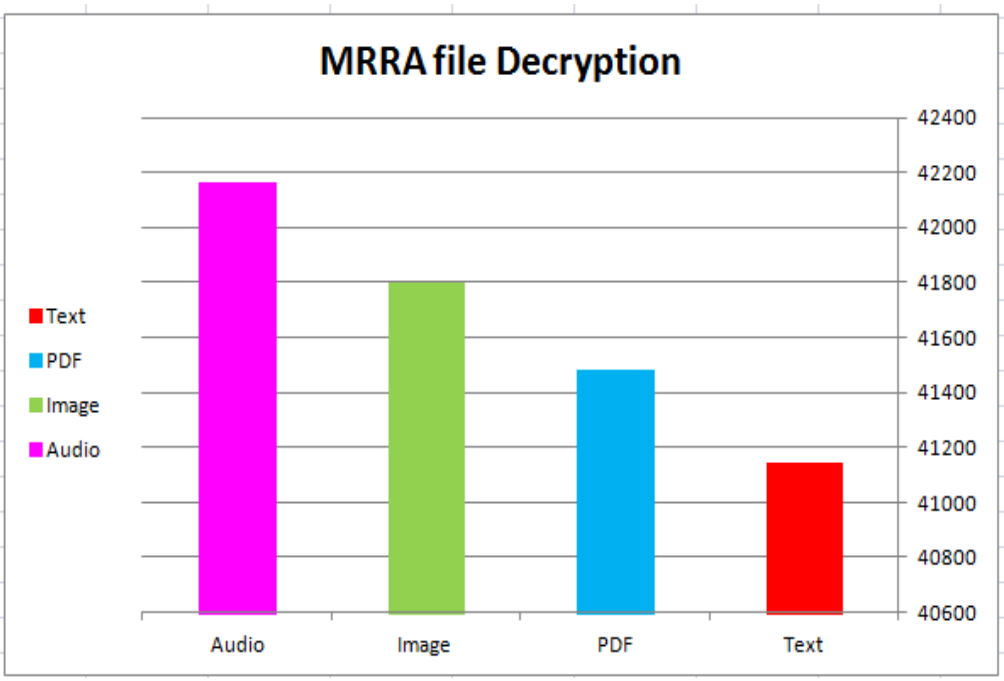

Figure 18. MRRA file decryption time in millisecond.

As it is implied on the figure above the decryption time of those sample files increased due to the increase of the file size of the samples from 100KB to 1MB. In addition to this as the file increases the decryption time gap went on increasing from file formats of text, PDF, image and audio files.

The encryption and decryption processes of the MRRA algorithm result can also be described in the table below.

Table 7. MRRA Encryption and Decryption time in millisecond.

Text Type	MRRA Encryption\Decryption With file size 100KB		MRRA Encryption\Decryption With file size 1MB	
	Encrypt	Decrypt	Encrypt	Decrypt
Text	4418.7	4018.2	45247.5	41810.9
PDF	4548.1	4051.3	46572.7	41485.3
Image	4852.1	4082.1	49685.5	41800.7
Audio	5417.5	42163.2	55475.2	42163.2

The table above generalize the entire experiment done in this thesis which includes the encryption and decryption processes of all the sample file formats of Text, PDF, Image and audio using both file sizes of 100KB and 1MB.

The figure below also demonstrate the distribution of the Encryption and decryption algorithm of the Modified Reduced Round Advanced Encryption Standard algorithm with a file size of 100KB.

The total experimental result of this thesis which refers to all sample file types of Text, PDF, Image and Audio formats which are tested using the modified AES algorithm of 8 round iteration with the same file size encrypted and decrypted by a single encryption key shows that there is a 4% execution time difference as demonstrated in a picture below.

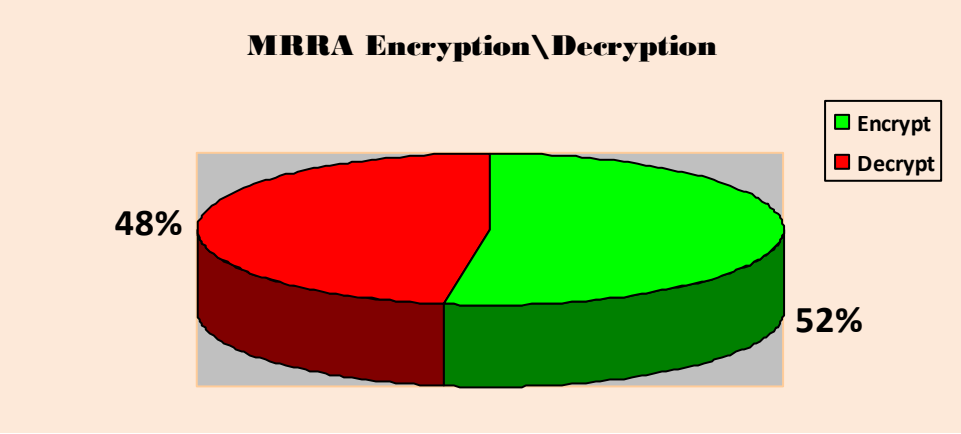

Figure 19. MRRA Encryption\ Decryption time percentage distribution.

Comparison between the standard AES algorithm in 10 round iteration and the MRRA with 8 round iteration are demonstrated in percentage as with the following tabular description.

Table 8. AES and MRRA comparison.

Standard AES and MRRA comparison in 100KB file size		
File Type	AES Encryption	MRRA Encryption
Text	4803.4	4418.7
PDF	4830	4548.1
Image	5127.3	4852.1
Audio	5721.6/ 20482.3	5417.5/19236.4

The above computation which are made to compare the file execution performance were measured finally between the standard AES algorithm with 10 round iteration and the modified AES algorithm focused by this thesis with 8 round iteration summarized by a table above for sample files with equal size of 100KB.

To briefly describe the above comparison the figure below can be drawn and demonstrate the execution time difference between the MRRA and standard AES.

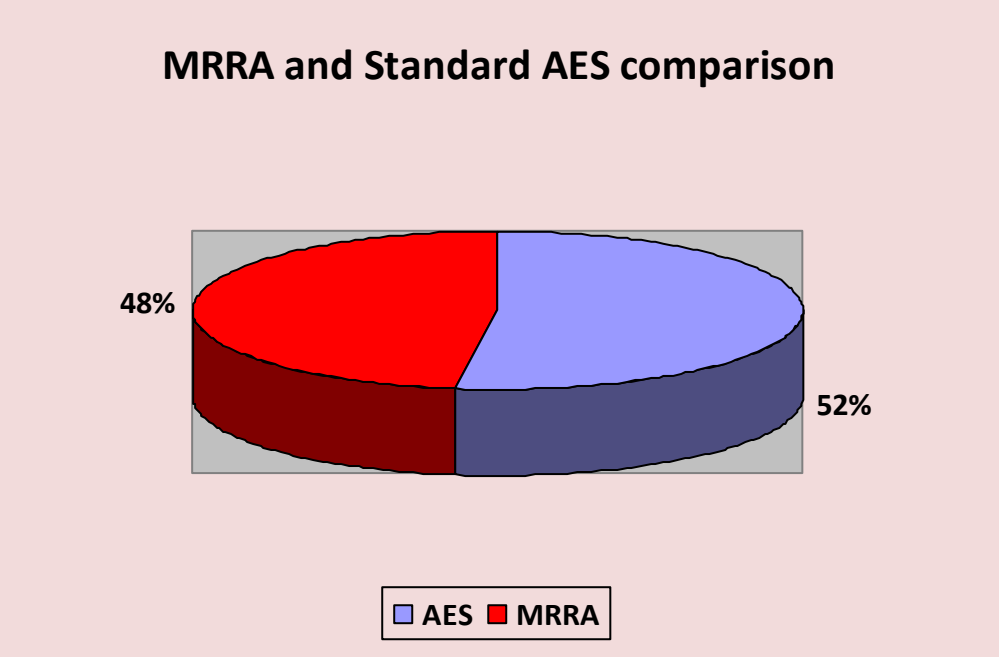

Figure 20. MRRA and Standard AES algorithm comparison.

The analysis done so far indicates that the MRRA able to reduce the execution time and increased the performances of the AES encryption\decryption algorithm through modifying the algorithm in terms of reducing the execution round from 10 to 8 round of iteration.

The result shown above in figure18 indicates the total execution time registered on testing all those different files with different data type in the same file size of 100KB resulted in an execution time of the standard AES algorithm of 52% with 10 rounds. The same experiment was done for the Modified Reduced Round AES algorithm (MRRA) and the result of the experiment indicated that the execution time were reduced to 48% with a total improvement of 2% were registered, therefore, it is possible to conclude that the MRRA has improved the standard AES algorithm performance which was done by another researcher with reduction of execution time with 6 round in an improvement of 1.29% was further improved by this thesis to 4% execution time improvement.

Table 9. MRRA Encryption in different file size

Text Type	MRRA Encryption With file size 100KB	MRRA Encryption With file size 1MB
	Encrypt	Encrypt
Text	4418.7	45247.5
PDF	4548.1	46572.7
Image	4852.1	49685.5
Audio	5417.5	55475.2

The above table shows the difference in the MRRA encryption process by taking the same files with same data types on performing encryption but with a different file size. The result reflect that as the file size increases from 100KB to 1MB there is a total difference of 177744.5 millisecond.

It is possible to conclude that as the file size increases the execution time also increased by large amount and hence we can conclude that files with large size requires higher execution time than those files with small file size.

4.8. Discussion

The problem identified on Business-to-Business e-commerce transactions in terms of delay and lose of confidentiality up on exchanging files and products on the online business transaction can be improved by the modified AES algorithm the primary focus of this research thesis.

The execution time performance improvement of 1.2% achieved with the 6 reduced round algorithm done by Edjie M [23] which has resulted with a security violation problem “Brouteforce" entails with loss of confidentiality on using the newly developed encryption algorithm were further improved by this research thesis with 8

reduced round iteration which were not created risk on its implementation registered an improvement of performance by 4%.

This research paper is different from other papers done in the area through modifying the AES standard algorithm from 10 iteration round to 8 iteration round which is not susceptible to a bruteforce attack and also this thesis has combined the MRRA with that of the MD5 hashing algorithm to ensure the key exchange problem of the standard AES algorithm.

The MRRA could be implemented when a valuable document and or information being transferred from the buyer to the seller and vice versa on the electronic market and also could be applied at time these valuable assets are stored within the computer system by encrypting and decrypting the files using the MRRA.

Chapter Five

5. Conclusion and Future Work

5.1. Conclusion

E-commerce is a new platform introduced to enable producers to produce and promote their products and services to the global market and attract potential buyers online without crossing boundaries and consume time and cost through traveling. The buyers also get benefit from this technology through buying any product being home from the global market with a fair price.

Cryptography is the science of protecting files and data items which are being stored within the database system and also those are being transmitted from one network environment to the other through various communication media.

To ensure the required security through cryptography there are two types of cryptography, the symmetric and the asymmetric cryptography.

One form of cryptographic transformation is symmetric encryption, where the transformation is reversible using a single secret key also called as a cipher key for both encryption and decryption. The other one uses two key for both encryption and decryption.

The focus of this thesis is on the symmetric cryptography and the standard algorithm used in the symmetric encryption is the Advanced Encryption Algorithm (AES). As various research papers indicated AES algorithm has registered fast execution time and performance than the other algorithms and also has ensured maximum security. This was one of the reason which initiated the researcher to focus on the AES algorithm and to enhance the performance of this algorithm further and to reduce the execution time.

One of the limitation of the AES algorithm is the exchange of key in between the source and the destination which hackers and crackers intercept the communication channel and got the encryption key and used for unauthorized access of valuable information and also used for theft of money.

The other focus of this thesis is to secure the key exchange between the buyer and the seller through using the MD5 hashing algorithm and encrypt the key and using the encrypted key for converting the plain text into cipher text and also exchange the hashed key with the receiver for decryption process.

|Therefore, four experiments were made first through firs selecting sample files with different data types of Text, PDF, Image and Audio files with 100KB file size both encryption and decryption are done using the modified AES algorithm and the result were measured and stored and next the same files are taken with a different file size of 1MB in order to test the impact of file size change in the encryption and decryption process through the modified AES algorithm and the result were measured and documented.

In the first measurement the sample files with a 100KB were encrypted using the standard AES and the modified AES algorithm using a Java Programming language in millisecond and the MRRA resulted with a total of 19,236.4 millisecond which is about 48%. The same files with the same size were tested in the case of the standard AES algorithm and the result found as about 29482.3millisecond which is about 52%. Therefore the result showed that there is an increase of performance in terms of execution time about 4% by the MRRA than that of the standard AES algorithm.

For the decryption process the same sample files with 100KB of each were measured in both algorithms. First MRRA has registered a total measurement of about 16,269.1 millisecond which is about 48% of the total execution time and that of the standard AES algorithm registered a total execution time of 16,465.3 millisecond which is about 50% of execution time. As it is shown here the decryption process in general registered less execution time than that of the standard execution time and the improvement of performance is only about 2% registered by the MRRA.

5.2 Recommendation

Maintaining strong security without compromising the performance of the algorithm is a must to be implemented in the electronic business to enhance the confidentiality of the system and also to increase the confidence of the buyers and the sellers to transact through this recent technology.

After analyzing the results gathered from the experiment and comparing with theoretical framework in various literatures and sound international practices the following recommendations are made with the objective to improve the e-commerce business transaction security:

- The AES standard algorithm shall be modified by reducing the number of iteration round from the standard 10 round to 8 round to enhance the performance of its execution time.
- We also recommend that to ensure the security of the standard AES algorithm basically the key exchange problem can be resolved by combining the Standard AES algorithm with the MD5 hashing algorithm.
- The buyer and the seller must exchange the key after being hashed by the MD5 hashing algorithm.
- Organizations must invest on maintaining such a research work which will further reduce the time and cost incurred on waiting until a complex encryption and decryption algorithms perform extended procedures.

5.3. Future Work

Researcher in the area can also further extend their work by enabling a mechanism that will enable the modified AES algorithm instead of counting the elapsed time by a manual watch counting it is better if the algorithm measure the time from the system and compute.

REFERENCES

1. Martin Kutz, " Introduction to E-commerce: Combining Business and Information Technology", 1st edition, Martin Kutz and bookboon.com, ISBN 978-87-403-1520 2, 2016.
2. Murphy, Ann and Murphy, et al "The Role of Cryptography in Security for Electronic Commerce," 2001. ITB Journal: Vol. 2: Iss. 1, Article 3.
3. Prof. Prathamesh Churi, "E-commerce Security with Secured Electronic Transaction protocol: A Survey and Implementation", ISSN No. 0976-5697, Volume 8, No. 8, September-October 2017.
4. M. Niranjanamurthy, C. Dharmendra, "*The study of E-Commerce Security Issues and Solutions"* International Journal of Advanced Research in Computer and Computer Engineering, vol.2, Issue 7, July 2013.
5. Asmaa Essam Alhibshi, "*Encryption Algorithms for Data Security in Local Area Network"*, Melbourne, Florida May, 2019.
6. Abdel-Karim Al Tamimi, "*Performance Analysis of Data Encryption Algorithms",* 2005.
7. Abraham Lemma," *Performance Analysis on the Implementation of Data Encryption Algorithms Used in Network Security*", ISSN: 2279 – 0764, Volume 04, July 2015.
8. H. Arif and S. Zarina, et al, "*An Efficient Secure Electronic Payment System for E-Commerce"*, UKM, Bangi 43600, Selangor, Malaysia, August 2020.
9. P. Priyadarshini, N. Prashant, et al, A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish, Procedia Computer Science 78 (2016) 617 – 624, December 2015.
10. Norman D. Jorstad, "Cryptographic Algorithm", Institute for Defense Analyses Science and Technology Division, January 1997.
11. Priyanka Walia, "*Implementation of New Modified MD5-512 bit Algorithm for Cryptography*", International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163, Volume 1 Issue 6 (July 2014).
12. Hanna Willa Dhany et al, "Encryption and Decryption using Password Based Encryption, MD5, and DES", International Conference on Public Policy, Social Computing and Development, (ICOPOSDev 2017).
13. Douglas Selent, "*ADVANCED ENCRYPTION STANDARD*", River Academic Journal, Volume6, Number2, 2010.
14. Anurag Rawa, et al, "Cryptography Algorithm", Journal of Analog and Digital Communications volume 2, 2019.

15. Abraham Lemma, "*A Review and Comparative Analysis of Various Encryption Algorithms"* International Journal of Security and its Applications 9(4):289-306 DOI:10.14257/ijsia.2015.9.4.27, 2015.

16. Abdul Monem S, "*Hybrid Model for Securing E-commerce Transaction*", International Journal of Advances in Engineering & Technology, ISSN: 2231-1963, Nov 2011.

17. Prakash Kuppuswamy, Y Saeed, "Securing E-Commerce Business Using Hybrid Combination Based on New Symmetric Key and RSA Algorithm", MIS Review Vol. 20, No. 1, September (2014), pp. 59-71.

18. Sidraah Matte, et al, "*Hybrid Model for Securing E-Commerce Transaction*" International Journal of Scientific & Engineering Research Volume 9, 25 ISSN 2229-5518, April-2018.

19. Jothina Mazarir, Clopas Kwenda, "*Hybrid Algorithm for E-commerce Applications*", International Journal of Research in IT and Management, Corpus ID: 125760098, 2016.

20. S Abdul Monem et al, "Hybrid Model for Securing E-commerce Transaction", International Journal of Advances in Engineering & Technology, ISSN: 2231-1963, Nov 2011.

21. Edjie M. De Los Reyes1, et al, "File encryption based on reduced-round AES with revised round keys and key schedule", Indonesian Journal of Electrical Engineering and Computer Science Vol. 16, No. 2, November 2019.

22. David Gstir, "Analysis of Recent Attacks on AES", Graz, 15 October 2012.

23. Edjie M. De Los Reyes et al, "File encryption based on reduced-round AES with revised Round keys and key schedule", Indonesian Journal of Electrical Engineering and Computer Science Vol. 16, No. 2, November 2019, pp. 897~905, May10, 2019.

24. Ejub Kajan, "The Security issues of B2B interactions"

https://www.researchgate.net/publication/265396662, Article · January 2006

Appendix

Sample Java code used in this thesis to test the selected sample file for both the encryption and decryption process

```
import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.security.InvalidKeyException;

import java.security.Key;

import java.security.NoSuchAlgorithmException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.crypto.BadPaddingException;

import javax.crypto.Cipher;

import javax.crypto.IllegalBlockSizeException;

import javax.crypto.NoSuchPaddingException;

import javax.crypto.spec.SecretKeySpec;


public class CryptoUtils {


/**

* A utility class that encrypts or decrypts a file.
```

```
* @author www.codejava.net
*
*/
private static final String ALGORITHM = "AES";
private static final String TRANSFORMATION = "AES";

public static void encrypt(String key, File inputFile, File outputFile)
throws Exception {
doCrypto(Cipher.ENCRYPT_MODE, key, inputFile, outputFile);
}

public static void decrypt(String key, File inputFile, File outputFile)
throws Exception {
doCrypto(Cipher.DECRYPT_MODE, key, inputFile, outputFile);
}

private static void doCrypto(int cipherMode, String key, File inputFile,
File outputFile) throws Exception {
try {
Key secretKey = new SecretKeySpec(key.getBytes(), ALGORITHM);
Cipher cipher = Cipher.getInstance(TRANSFORMATION);
cipher.init(cipherMode, secretKey);

FileInputStream inputStream = new FileInputStream(inputFile);
```

```
byte[] inputBytes = new byte[(int) inputFile.length()];
inputStream.read(inputBytes);

byte[] outputBytes = cipher.doFinal(inputBytes);

FileOutputStream outputStream = new FileOutputStream(outputFile);
outputStream.write(outputBytes);

inputStream.close();
outputStream.close();

} catch (NoSuchPaddingException | NoSuchAlgorithmException
| InvalidKeyException | BadPaddingException
| IllegalBlockSizeException | IOException ex) {
throw new Exception("Error encrypting/decrypting file", ex);
}
}
public static void main(String[] args) {
String key = "This is a secret";
File inputFile = new File("C:\\Users\\TOSHIBA\\Desktop\\Hybrid RSA.pdf");
File encrypted = new File ("enc.txt");
File decrypted = new File ("dyc.txt");
try {
CryptoUtils.encrypt (key, inputFile, encrypted);
```

```
} catch (Exception ex) {
Logger.getLogger (CryptoUtils.class.getName()).log(Level.SERVER, null,ex);
}
try {
CryptoUtils.decrypt(key, encrypted, decrypted);
} catch (Exception ex) {
Logger.getLogger (CryptoUtils.class.getName ()).log (Level.SERVER, null, ex);
}

}

}
```