جامعة الشرق الأوسط
MIDDLE EAST UNIVERSITY

Performance Analysis for Hashing Over Encrypted Data Techniques

تحليل الأداء لتقنيات تجزئة البيانات المشفرة

Prepared by

Esra'a Sameer Al-Rawashdeh

Supervisor

Prof. Ahmad K.A. Kayed

A Thesis Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Information Systems

Department of Computer Information Systems

Faculty of Information Technology

Middle East University

Amman, Jordan

May, 2016

Middle East University

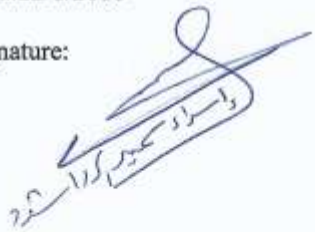
Authorization Statement

I, Esra'a Sameer Al-Rawashdeh, authorize Middle East University to supply hardcopies and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulation.

Name: Esra'a Sameer al-rawashdeh.

Data:22/8/2016

Signature:



إقرار تفويض

أنا اسراء سمير الرواشدة، أفوض جامعة الشرق الأوسط للدارسات العليا بتزويد نسخ من رسالتي ورقيا و الكترونيا للمكتبات أو المنظمات أو الهيئات و المؤسسات المعنية بالأبحاث و الدارسات العليا عند طلبها.

الأسم: إسراء سمير فلاح الرواشدة.

التاريخ: 2016/8/22

التوقيع:

Examination Committee Decision

This is to certify that the thesis entitled "Performance Analysis for Hashing over Encrypted Data Techniques" was successfully defended and approved on 24/5/2016.

Examination Committee Members **signature**

(*Supervisor*)

Prof. Ahmad Keyed

Associate Professor

Dean Faculty of IT

Middle East University

(*Head of the Committee and Internal Committee Members*)

Dr- sharefa murad

18/8/2016

(*External committee members*)

Dr- shadi aljoarnh

Acknowledgements

“In the name of Allah the Most Gracious the Most Merciful”. I would like to thank and praise my God "Allah” for everything he has given me. He has given my guidance, my health, my study, for smoothing my research task, and for helping me to worked this performance and achievement.

I offer all thank and love to my parents (Sameer Falah Al-Rawashdeh and Ghadaa Attalla Al-Rawashdeh) for their encouragement and their moral supports during my study.

I would like also to express my deepest gratitude and appreciation to my thesis supervisor (Prof. Ahmad Keyed) for his effort, helpful instructions, and guidance, support, enthusiasm and inspiration during the work of my thesis. I really thank him for his comments which enriched the quality of this work.

My special thanks and appreciation also extended to the committee members for taking part in the discussion of this thesis and for their valuable comments and suggestions. My special thanks also go to all my friends at Middle East University who somehow helped me. I would like to express my deepest love and gratitude to myhusband Dr. Hisham Al-kasasbeh, my brothers (Anas& Muath) and sisters (Shahed& Anwar) for their unlimited encouragement, valuable support and patience throughout my study.

بسم الله الرحمن الرحيم

"وقل ربي زدني علما"

Dedication

I dedicate this work to my father, my mother, my husband, my uncles, my aunts, my brothers and sisters. The completion of this work would not have been possible without their support. I also, dedicated to all my teachers, Specially Dr. Arwa Aldbayat and Dr. Nedal Al-Ameren.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

RSA	Ron Rivest, AdiShamir and LeonardAdleman
NIST:	U.S. National Institute of Standards andTechnology.
MD5	message-digest algorithm
SHA	Security hashing application
DES	Data Encryption Standard.
Enc	Encryption and decryption
3DES	Triple encryption standard
CC	CloudComputing.
CPU	Central ProcessingUnit
Enc	Encryption Algorithm.
OS	OperatingSystem
PaaS	Platform as a Service.
SaaS:	Service as aService
IaaS	Infrastructure as a Service.
DaaS	Database as service.
ISPs	Internet Service Providers

Performance Analysis for Hashing Over Encrypted Data Techniques

Prepared by

Esra'a Sameer Al-Rawashdeh

Supervisor

Dr.Ahmad K.A. Kayed

Abstract

This study is based on information security principles such as confidentiality which can be achieved by encryption. However, using these principles for encryption only as an example led to numerous obstacles. The hashing technique has been used to index Database. The encrypted hash will cause a problem for indexing database. Thus, combination techniques are applied to solve these issues. Popular cryptographic algorithms such as Data Encryptions Standard (DES) and Triple DES have been used in this thesis.

This study looks into the latest solution proposed by these cryptographic algorithms, and their parameters to find the best efficiency of security enable hashing over encrypted data. These parameters are encryption and Hashing Techniques and plaintext size. This study took two approaches:

Encryption Hashing: this has been done through entering plaintext into the hash and calculating execution time and CPU time before the encryption process, then after that, the same process for the encryption algorithm. The second approach is Hashing over Encrypted Data: This has been done through entering the plaintext into a hash, after that combining the hash results with the plaintext, in such way that we calculate the processing time and CPU time before encryption then passing it into encryption algorithm.

The study compares between these two techniques by using the following parameters: the size of the plaintext, various hashing and encryption cryptographic algorithms. The target from this to evaluate each parameter effectiveness on performance.

Depending on the previous statement, these parameters have been studied and tested in order to achieve the high level of security and efficiency. Furthermore, we used random plaintext with different sizes. Thus, the simulation executed upon these parameters in order to achieve the expected results, the evaluation is done through time consuming and CPU time before and after first and second encryption process that took phase.

In all cases, this thesis finds out that DES and MD5 are the best encryption and hashing techniques with results to performance.

Keywords: Cloud computing, cryptography, encryption Algorithm, and hashing algorithm.

تحليل الأداء لتقنيات تجزئة البيانات المشفرة

إعداد

اسراء سمير الرواشدة

اشراف

د. أحمد الكايد

المُلخص

يعد امن البيانات من اهم القضايا التي تم طرحها المعلومات مثل السرية التي حيث يتم معالجته عن طريق التشفير. تشفير البيانات يؤدي الى مشكله جديدة. لقد استخدمت تقنيات التجزئة على بيانات مؤشر. التجزئة المشفرة تسبب مشاكل للبيانات الفهرسة. وبالتالي، يتم تطبيق مجموعة من تقنيات لحل هذه القضايا. وقد استخدمت تشفيرات البيانات الموحدة (DES) و ثلاثي البيانات الموحدة (Triple DES)هي هذه الاطروحة.

هذه الدراسة احدثت حل مقترح من قبل هذه خوارزميات التشفير، والمعلمات من أجل العثور على أفضل كفاءة الأمان ولتمكين تجزئة على البيانات المشفرة. هذه المعايير هي: التشفير والثرم تقنيات، حجم النص. أخذت هذه الدراسة منهجين:

تشفير التجزئة وهي إدخال نص الى الهاش وثم حساب وقت المنقضي ووقت CPU قبل التشفير ومن ثم تمريرها داخل خواروميات تشفير وثم حساب الوقت المنقضي لتشفير نتائج الهاش و وقت CPU . المنهجية الثانية هي تجزئه البيانات ادخال نص الى الهاش ومن ثم دمج نتائج الهاش مع النصوص، بحيث يتم حساب وقت العملية و وقت CPU قبل تشفير ومن ثم تمريرها داخل خوارزميات تشفير.

دراسة المقارنة بين الخطوتين مع استخدام المعاملات التالية: عدة احجام من نصوص، عدة خوارزميات الهاش و عدة خوارزميات التشفير. والهدف هو دراسة تأثير تلك المعاملات على مستوى الاداء.

وبناء على ما سبق, تم دراسة تلك المعاملات من اجل تحديد ايها تحقق مستوى امن مرتفع وافضل اداء. وفي هذه الأطروحة تم استخدام نصوص بيانات عشوائية ذات احجام مختلفة. وتم تشغيل البرنامج بناء على المعاملات للحصول على النتائج المرجوة والمقارنة بينهم. عن طريق حساب الوقت المنقضي و وقت CPU قبل تشفير وبعد تشفير في العملية الاولى والثانية.

في جميع الحالات، هذه الأطروحة تجد أن DES وMD5 هي أفضل النتائج لتقنيات التشفير والتجزئة بالنسبة للأداء.

الكلمات المفتاحية: الحوسبة السحابية، التشفير، خوارزميات التشفير وخوارزميات التجزئة.

Chapter One

Introduction

1.1 Overview:

In this chapter, essential basic information and the scope of the thesis is presented as an introduction. After that an idea about the research problem is given and how it has been treated, thesis questions, research objectives, motivation and related work.

1.2 Introduction:

Due to the spread of wireless applications, the user confidential data are being stored on the internet. Thus, security becomes a great concern. Therefore, many of the research give an importance of cloud computing. In this environment, people send and store their data in a location over the internet. Studies in this field show an increasing in the rate of attacks and attempted attack either to obtain information or destroyed it. The security problem is eminent with cloud computing environment. Totally secure and protected data exchange environment is still the objective which has not been accomplished, and it needs a large number of studies and research.

In the recent years, Cloud computing has appeared in our lives by work online to be new model wide range and platform in business. cloud computing contains a set of applications that use the powerful servers and data centers that host applications needed by the user and can be obtained via the Internet (Vijayaraj& Ram , 2011).

These applications in cloud computing to users and business organizations, used as a solution for services in the field of media and information technology (IT) (Nazir et al., 2015).

There are many definitions mentioned in various literature, the standard definition from NIST (National Institute of Standards and Technology)defines the cloud computing: “Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network

access to a shared pool of configurable computing resources(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned (Mell, et al., 2011).

Another definition of Cloud computing is continuously developing and there are many major providers such as Amazon, Yahoo, and Google who are providing services such as software as services, platform as services and infrastructure as services (Rana and Josh, 2012).

Figure 1.1 show cloud computing environment.

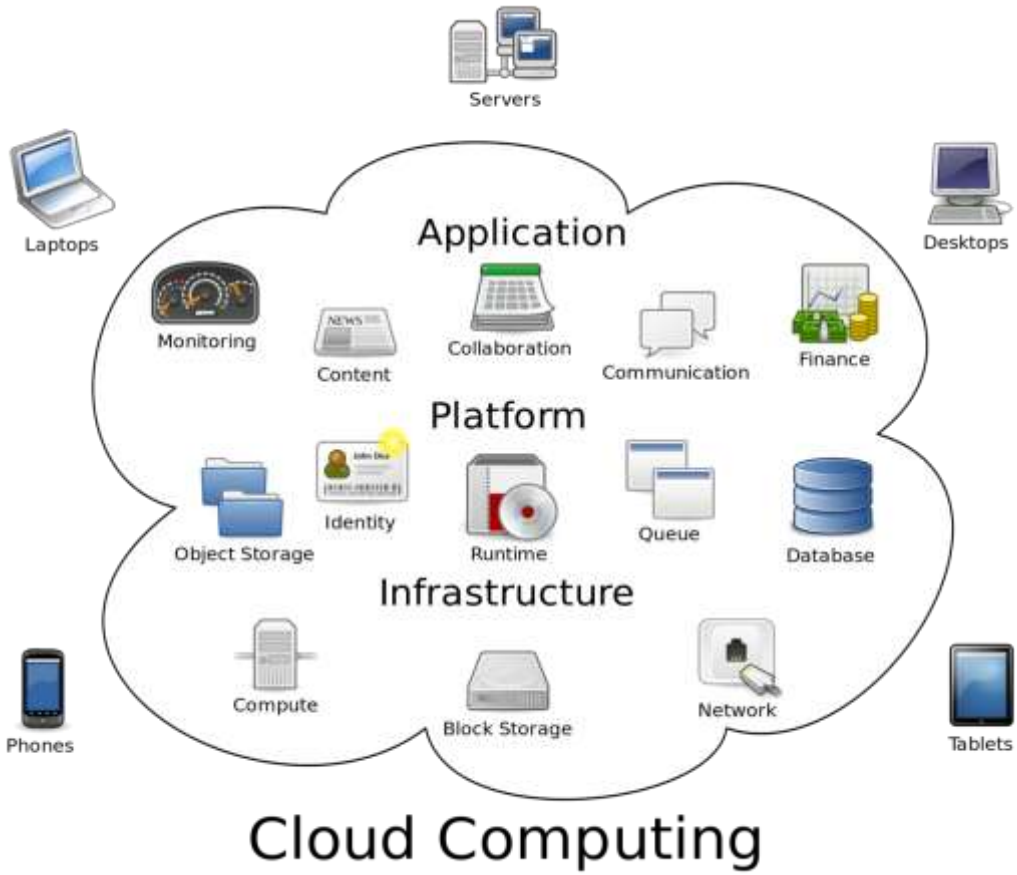


Figure 1.1 the cloud computing environment (Nazir et al., 2015).

Nowadays, the Cloud has been rated within several models based on the type of service provided to its users. Cloud computing services can be used in a private, public, community and hybrid preparation. Although the Cloud Computing has several benefits,

but it has some challenges (Singla& Singh, 2013).Subsequent chapters present a detailed examination of that challenge.

Some person uses the cloud everyday on the internet without knowing that. But users cannot know where their data are being kept. As mentioned before, the data transmit over the internet and it is stored on remote locations. In addition, the cloud provides services to multiple customers simultaneously. All of this leads to raising the problem of security in cloud computing and other problems that make users fear to deal with cloud computing. (Apostuet al., 2013).

In the line of these developments on cloud computing security issues, the attacker takes the main concern. Attackers gaining unauthorized access to the confidential data stored in the cloud. Therefore, employ a variety of techniques to gain access to clouds without legal authorization as shown in figure 1.2 According to DataLossDB, there were 1279 data breach incidents during the first eight months of 2014, compared to 1472 incidents during the entire year of 2015**(**Chou. 2013).

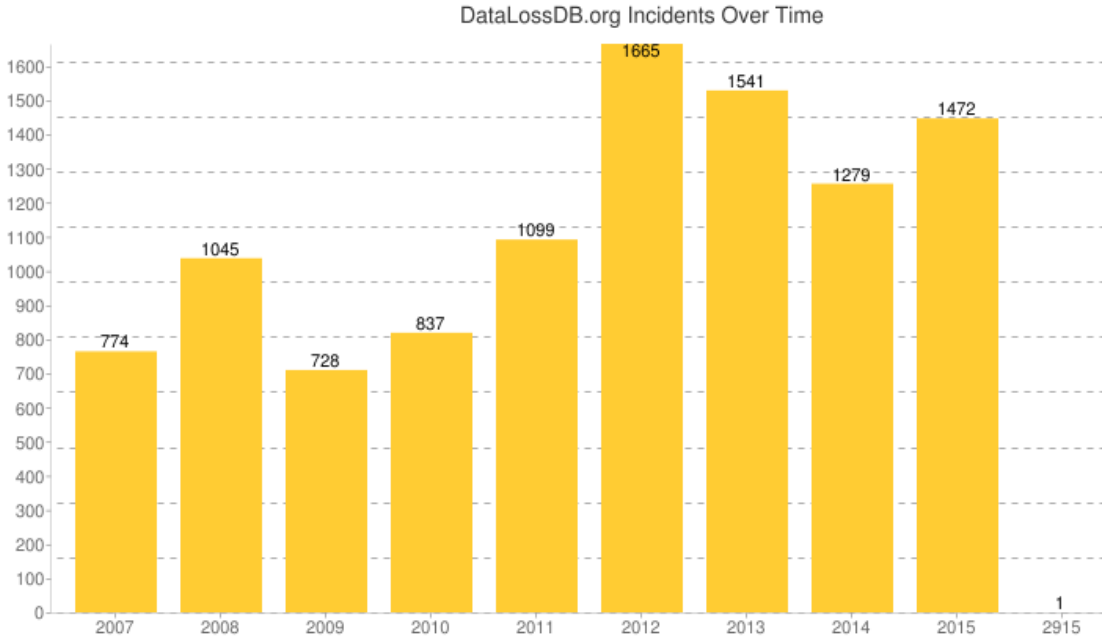


Figure 1.2: data breach incidents according DataLossDB (online DataLossDB Open Security Foundation).

Exchanging of information requires more security and reduction whether in the space need for data storage and the time for data send. There are many researchers and articles are working to solve these challenges to gain user trust in dealing with cloud computing (Bisong and Rahman, 2011). One of a solution is to reduce the challenges of Cloud Computing using techniques based on virtualization (Koganti et al., 2013). Usually, in the cloud computing virtualization plays a big role. Virtualization is software run on one device multiple operating system or multiple applications, independently of each other (Teeba, and EL hajii, 2013). Another solution to achieve the security and privacy within the CC is the encryption (Prasanthi, et al., 2012).Internet use increasing rapidly, so there is need secure on data run on the internet using several services. All there provide security for data on the internet by using different encryption algorithm. At this research, many well-known encryption algorithms are used; they are DES and 3DES.There are two types of Encryption Technique that preserve operations on data: Homomorphic encryption technique and Order Preserving Encryption Technique (OPE). The process plaintexts convert to ciphertext to provide high security with an attack called encryption (Kumar et al., 2012).

Encryption has been used for centuries to protect political and military secrets. In a history of encryption, the security issue has taken a primary seat by using symmetric (private key). In mid-70 Diffie and Hellman, were worked trend changers for the advent of public key (asymmetric Encryption). Figure 2.1 show types of encryption.

Shi, Ma, Cote, and Wang, (2012**)** explained a number of related concepts like Digital Signatures, integrity, and Authentication. These very important because use in cryptography named cryptography hash functions.

Encryption algorithm performance impacts with the equipment detail such as core. Numerous PCs today have multi-center processors, which means the CPU contains more than one center.

Nowadays, the Communication has a complete change world. Communication relay needs to be more secure to attack activities. Encryption hash functions that are used to provide data security and build integrity checking techniques (Raj et al., 2013). Subsequent chapters present a detailed examination of that cryptography and cryptography hash functions.

Conducting extensive research has been passing in the field of Cryptographic Hash Functions.Hash Functions are being created from existing primitives like Block ciphers. As well as using special family like MD (MD5) (Xie1et al., 2013). And SHA (1&265) (Romine. 2012), (Roldan et al., 2012) and (Gupta and Kumar. 2014).

Company NIST has stated SHA-1 exposure to attacks from the attacker, therefore tended to use the same sizes as big retail SHA-256 (NIST, 2004).

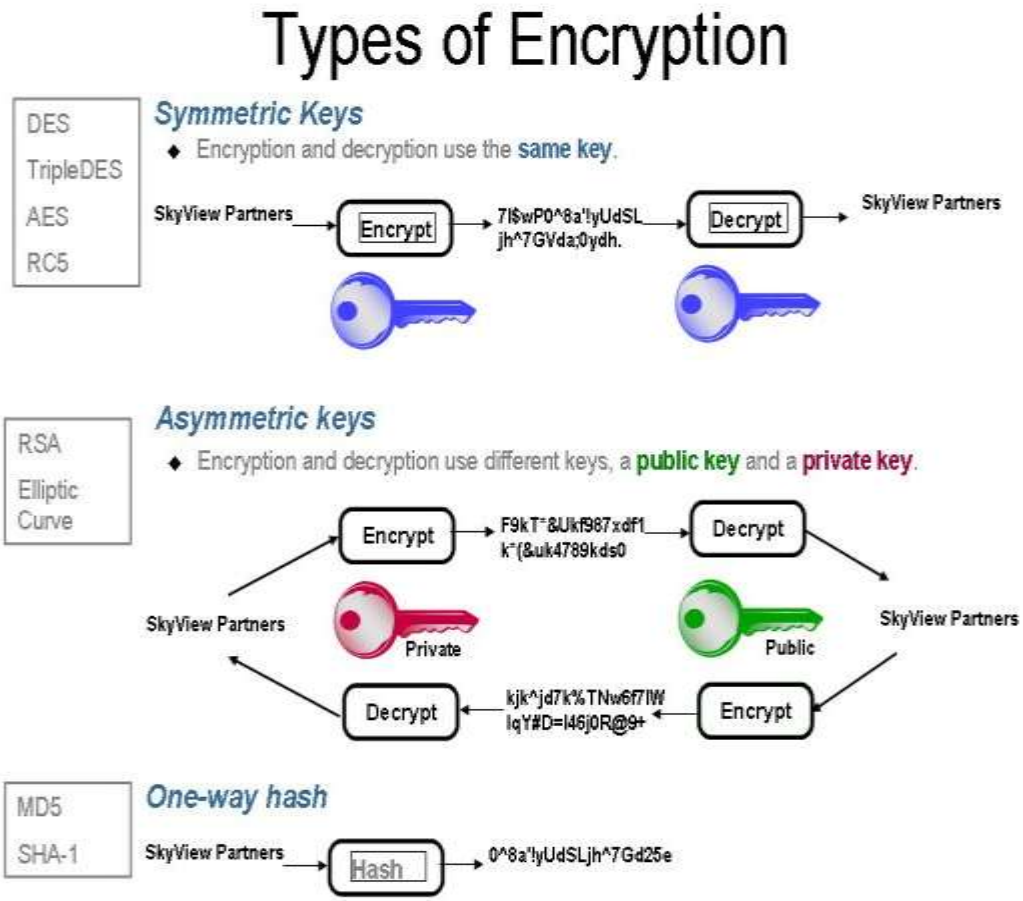


Figure 1.3 Types of encryption (Woodbury. 2007)

At this research, many well-known hashing algorithms are used; they are MD5, SHA-1, and SHA-256.

The main important points in the database are indexing and hashing. This thesis compares among encryption hashing and hashing over encrypted data. This thesis studied among steps by used encryption function and hash function with the following parameters: the key sizes and data sizes find by calculated time and CPU. The point of these parameters is decided the loss of performance of the two steps. The key sizes have been used in the encryption technique to study the effect of these key sizes on the performance. The aim of this thesis is to find the better encryption algorithm, and hashing

algorithms as a performance by execution time, percentage CPU.

This proposition will concentrate a set of the models or frameworks proposed as of late in the field of information encryption, to take advantages from these researches to build our special model.

1.3 Problem Statement:

People send and store their data in a location over the internet and they do not want to be controlled by anyone. Lately, many types of research have been carried out in cryptography, leading to a converted data from readable to unreadable data called encryption. Encrypting data creates new problems for indexing and hashing. This research will focus on hashing issues. Mainly, several encryption techniques with several keys may give different hashing value for the same text. One solution for that is to calculate the hash before encryption. But if anyone knows the hash function this will leak some knowledge about the data by brute force. Several encrypted hashing functions can be used to reduce this leak. Solving this by merging value hash with the text before encryption then pass it to the server side. This research will investigate the effect of hashing and encryption process, with several parameters to study their effect on performance. These parameters are encryption algorithm, hashing algorithm, and text size. This research finds the optimal values of these parameters to find the best performance (CPU speed).

I1.4 Research Questions

Problem will be accomplished by answering the following questions:

- What are the main parameters that effect hashing over encrypted data?
- How can we measure the effect of parameters for hashing over encrypted data?

The above questions have been extended to investigate the effect of these parameters on security.

1.5 Aims and objective:

This research work is to understand and identify the compare among encryption hashing and hashing over encrypted data with use many parameters: data size, encryption algorithms and hashing algorithms. The main aims of this research, effects parameters on performance level.

The main goals of this research are:

- Identify the main issues related to hashing and encryption.
- That the hashing techniques will affect the performance of encryption techniques.

1.6 Significance of the Study:

The importance of this study lies in the ability of analysis the main parameter that effect hashing over encrypted data. This research will compare encryption hashing and hashing over encrypted data. The outcomes of this research are important since it will define which hashing goes better with several encryption techniques.

1.7 Motivation:

The primary motivation to do this project is that people focused on use the internet through the focused to the better security provided for users on the cloud. Affecting trust users on our internet. Recently, must be maintaining public data and all users who enter the internet.

Cloud computing establishment has resulted in creating huge data centers around the world containing thousands of devices. Thus, huge amounts of data are entered to the data center. Nowadays, the researchers are giving greater attention to security it has becometheir priority because of the high attacker .Additionally, it is a searcher first priority to finding the best techniques for search data and security of data.

Brute force on data in data centers will continue to increase rapidly unless effective techniques are applied. Performance outcomes of encryption hashing are that it is a low number of attackers on the data center.

Security of course is a major concern in the datacenter and it is being in this research compared encryption .Encryption techniques will be tacked, these techniques are (DES and 3DES) and hashing techniques; while these techniques are (MD5, SHA-1, and SHA-256). This can be done by percentage usage of CPU and execution time of encryption); compare between the result hashing encryption and hashing over encrypted data.

1.8 Methodology

This research will build several experimentations to find the best algorithms by calculated percentage usage of CPU and execution time of encryption. This methodology mainly based on studying and implementing encryption (encryption algorithm and hash algorithm). And monitor the performance (percentage usage of CPU and execution time of encryption) of the algorithm with several parameters. The proposed solution will use a quantitative methodology in building these experiments with different data that affect the performance of the framework. We built the programme on C sharp and assigning the proposed work on C sharp in order to run. Additionally, number of issues to will be addressed, such as:

- Discovery, rating, and analyze the research in the encryption hashing and hashing over encrypted data computing to get an encryption and hashing of the existing techniques.
- Conduct analysis of algorithms on C sharp to get performance (percentage usage of CPU and execution time of encryption).
- Implementation and evaluation the proposed operation.
- An action a set for the algorithms in program C sharp to compute the best

algorithm by percentage usage of CPU and execution time of encryption

- The conclusion that shows and explains the process and the results.

The methodology will contain the following steps:

1.8.1 Studying Phase:

The researcher will define the main techniques that will use for studying the best algorithms (encryption and hashing) on encryption hashing and hashing over encrypted data by (percentage usage of CPU and elapsed time of encryption) before and after encryption of the system.

1.8.2 Implementing Phase:

- Collecting data (plain text).
- Collecting many hash functions.
- Collecting some encryption techniques (DES and 3DES).
- Collecting several hashing techniques (MD5, SHA-1, and SHA-256).
- Run many experiments for several (hash function, an encryption technique, and

- compute time and CPU.
- The result.

1.9 Thesis Layout:

This thesis is divided into five chapters: -

- **Chapter one:** contains general concepts of this research that include the overview introduction cloud computing, statement of a problem, objectives, motivation, and methodology, proposed of work and finally it presents thesis layout.

- **Chapter Two:** reviews introduction, cloud computing overview, also presents an overview of cryptography algorithms, gives an Order-preserving encryption scheme and homomorphic overview then overview the hashing, overview indexing, overview encryption hashing/ algorithm and finally it lists the literature review.

- **Chapter Three:** reviews explain the methodology in details. This chapter includes the combination of encryption hashing and hashing over encrypted data.

- **Chapter Four:** experiment design presents the experiment result the experiments were explained result on the performance and security profit

- Finally, **Chapter Five** contains the conclusions of this thesis and suggests future work.

CHAPTER TWO

Background and Literature Review

2.1 Back Ground:

This chapter provides a brief overview of the background of cloud computing, cryptography, indexing, and hashing.

2.2 What is Cloud Computing?

Cloud computing refers to all application on the internet; where hardware and software in data center provide these applications. Application on the cloud contain server (hosts) and it uses high or huge data.

2.2.1 Before Cloud Computing

Much sooner than the term cloud computing was authored, programming suppliers were giving administrations to their clients from remote servers by means of web-empowered PCs. This was called Application Service Provision (ASP) and was the first stage. However, the ASP model ultimately was an experiment that failed because it involved more complicated initial installation; involved with today on-demand cloud services. It originated as a means of providing software at one to one basis rather than at the one to many major of cloud computing (Gorelik, 2013).

2.2.2 Cloud Computing Overview

Gonzlalez N. et al. (2012) defined the Cloud Computing is all services provides over the internet; Cloud services allow users use computer resources and access data anytime. The data is not stored on your desktop or your device but is located far away in the cloud.

2.2.3 Benefits of Cloud Computing:

Singh et al. (2014) described Cloud Computing with many benefits, which suit the definition explained above:

- **Resource pooling:** the cloud providers are doing pooled resource on cloud and shared this requirements or resource between users.

- **Rapid elasticity:** the cloud computing provides elasticity of requirements very speedily in a matter of the minute.

- **Broad network access:** the users can access to requirements on cloud by standard network
- **On-demand self-service:** can be providing requirements and resource every time.
- **Measured service:** CPU can measure many charges such as usage of CPU, memory, Network bandwidth.

Due to Cloud computing, the user can access and use of computing infrastructure according to required it on demand and reduce cost (buy the computer, provide memory and place). As the time passes, must be the interest of cloud computing raises between companies.

2.2.4 Entities of Cloud Computing

The Nazir et al. presented important entities used in cloud computing. Four major entities have been identified who work out cloud computing activities. Figure 2.1 shown the actors according to the NIST such as:

- **Cloud Providers service:** Cloud Providers or end user is control access to cloud service. It includes internet service providers (ISPs) and telecommunication organizations. Also, include data center to host private cloud and many services such as SaaS, PaaS, and IaaS to their user. Offering service on cloud and allow users to use them without the need to hardware and software highly cost. Example provide provider on cloud IBM, Google.
- **Cloud Service Brokers:** Responsible for combining cloud services. The userdetermines cloud computing service on cloud providers by adding many

services together on cloud providers. Also, Cloud Service Brokers involve technology consultants, business organizations, registered brokers, and agents.

- **Cloud Consumers:** User can use cloud customer by buy and use service from the cloud provider. Also, vendor and broker can exchange anything with another broker and vendor in another cloud provider.
- **Cloud Resellers:** Resellers is the basic stone on the cloud provider that increases their business through the cloud market. Also, the reseller can choose companies to promote user buy their product on the internet (Nazir et al., 2015).

- **Cloud Carrier:** Allow transport between Cloud Providers to Cloud Consumers through network and telecommunication. The user can require encrypted connection.

- **Cloud Auditor:** In cloud auditor, the cloud providers evaluate and audit for privacy, security, and performance. Auditing is very important to ensure the vendor work as expected; should include a contractual to assess the security of cloud provider.

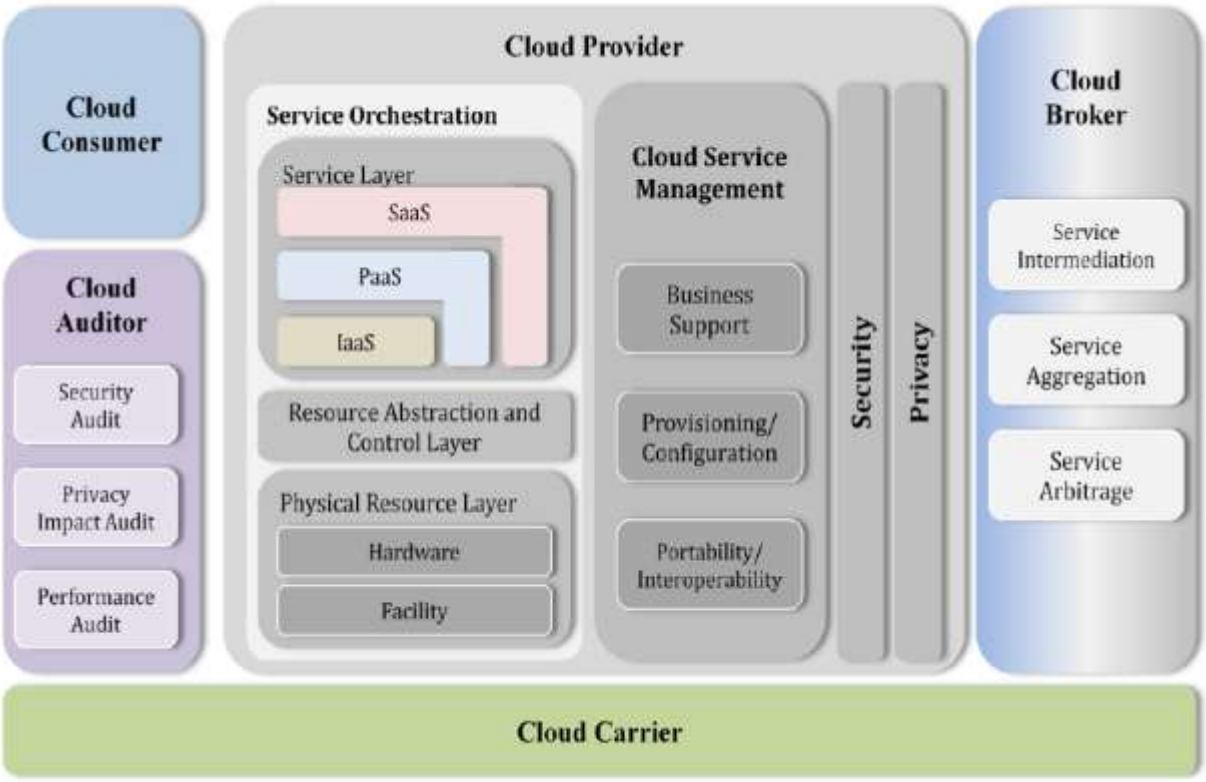


Figure 2.1: NIST Cloud Computing Entities Model (NIST, 2011).

2.2.5 Service model of cloud computing:

The cloud consumer needs many different kinds of services, services are different and are divided into three layer classifications depending on NIST, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). In each kind of the service hosted and accessed over the internet. In figure 2.2 shown layer architecture (service model) of cloud computing.

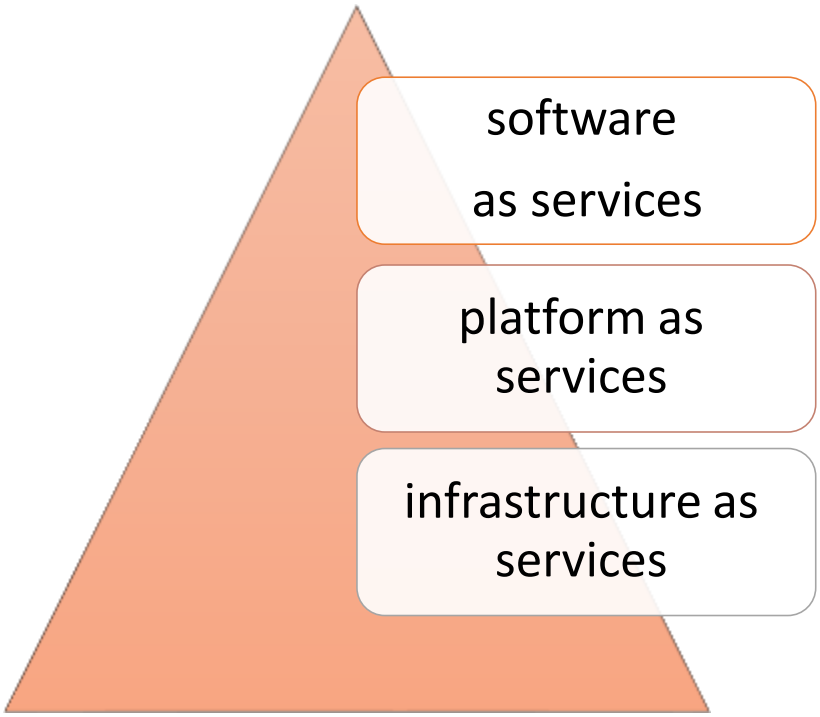


Figure 2.2 shows layer architecture (service model) of cloud computing (Kaur &Mahajan, 2013).

- **Software as a Service:** Represents the top layer, SaaS is the gate represented the interaction between the user and the service; it helps the user to access his requirements and to allow him to use the applications by hosting on remote servers. SaaS referred to the provision of application in the cloud (Rana& Joshi, 2012) &(Kaur & Mahajan, 2013). Forthe example of SaaS is Microsoft Office 2010 it provides directly use for browser-based applications (Apostu et al., 2013).
- **Platform as a Service:** PaaS which means allowed the deployment of Software and applications on the cloud and it allows users to access them, but do not publish operating systems and Networks (Rana, et al. 2012). Because they provide runtime (database, application server created on using programming languages) such as Google's App Engine (Apostu et al., 2013). The vendor uses

to describe their products (Kaur & Mahajan, 2013).

- **infrastructure as a Service:** Finally, IaaS used to describe services hosted over the Internet (Rana, et al. 2012); this is virtual can provide memory and computing power, storage space and network capacity which enable customer run software including application and operating system such as; Amazon WS service (Apostu et al .,2013).

Today, several pieces of research reported a new cloud service model called **Database as a Service (DBaaS) or storage as services** defined administration of cloud computing which permits information proprietor to move information from local frameworks to the cloud for reducing capital spending such as Amazon Relational Database Service (RDS) and Microsoft SQL Azure (Gawande and Kapse., 2014). So it managed by administrators and allow a user to retrieve, upload and delete the data on the internet.

DBaaS on the cloud are designed to meet several essential requirements of applications: manageability, scalability or elasticity, availability, and low latency. With a DBaaS, the application developers will not need to be database specialists; neither should they need to contract a database manager (DBM) to keep the database (Wu et al., 2010).

The benefit of using cloud DaaS is adding of additional nodes when required online, and increasing the performance of the database. With DaaS the database must get it all the time and where so that the user can get the data whenever he requires, then; the cloud database must reduce the costs as well. Data privacy is rated an important issue for any database user. In cloud computing, the data must be protected because it will be shared with any person (Al Shehri., 2013).

The cloud computing raises numerous security concerns like network security, interfaces, data security, virtualization governance, compliance, and legal issues (Makkar and Rajput. 2013). Thus, Data security means CIA (Confidentiality, integrity, and data availability) (Checker, et al. 2012):

- **Confidentiality**

  The confidentiality means that only the authorized person can read the data and that is realized by encryption such e-voting.

  In cloud computing to achieve confidentiality, we must choose the appropriate solution (Cryptography, Redundancy, and Disposal) (Gonzlalez et al. 2012).

- **Integrity**

  The integrity means the data cannot be modified and if a third part interceptor accesses the data he cannot change it and that is done using hashing.

- **Availability**

Availability means the data must be available to the user all the time he needs it.

A Database has turned into a critical part of cloud computing. It can be defined as any user can be accessed through the internet to Cloud database service provider and it can determine for demand which he needs (Al Shehri., 2013).

The challenges of DaaS do not address multi-tenancy, elastic scalability, and database privacy. Must be overcome before outsourcing database software becomes charming to many users. The data must be encrypted before sending to the cloud to prevent attacker entry and change to the original data.

2.3 Asymmetric and Symmetric Cipher Model:

The word cryptography comes from the Greek words κρυπτο (Al-Vahed& Sahhavi., 2011).Cryptography is referred as (security protection or secret on data). Therefore,

there are many ways used to achieving this goal, such as Encryption. Encryption is a secure coding technique in a given document, whose purpose is to reduce both the space requirements, time and save sensitive data. It is a model from an input plain text (string of symbols and arithmetic) coding to achieve security, is produced private key

Encryption, which are the elements that turn a general encryption algorithm into a specific method of encryption (Singh & Gilhotra., 2011).While cryptographers invent private codes, cryptanalysts break these codes. Table 2.1 shows major terminology of encryption.

Table 2.1 shows major terminology of encryption

Terminology	Description
Plain text	The original message
Cipher text	Coded message(encrypted message)
Cipher	algorithm for transforming plaintext to cipher text
Encryption	The process of transforming plaintext into cipher text.
Secret key	One key used for encryption and decryption (symmetric key).
Public key	At least Two key use encryption and Decryption (a symmetric key).
Encryption algorithm	Executes encryption by input and key
Cryptography	Topics for encryption or Science of studying cryptographic system
Cryptology	both cryptography and cryptanalysis
Stream ciphers	encrypts one bit or one byte of plaintext data
Block ciphers	encrypts a block of plaintext data
Cryptanalysis	The study method of decryption without knowing key
Decryption	The process of transforming cipher text into plaintext.

There are two types of encryption techniques. The first type is using the same key for encryption and decryption this is type is called symmetric. While other techniques is using different keys for encryption and decryption this is called asymmetric. In the following we will explain these two types (Gupta &Kumar, 2014).

2.3.1 Asymmetric Cipher Model

Asymmetric key also called public key is cryptography in which two keys one key is used to encrypt another key used decrypt a plain text. See figure2.3 there are six elements; plaintext, encryption algorithm, Public and private key, Ciphertext, and decryption algorithm (Sing and Gilhotra., 2011).If a sending user encrypt the plaintext with the public key and again with the secret private key. Then decrypt same operations (Sangwan, 2012) for example RSA (Shankar & Akshaya, 2014).

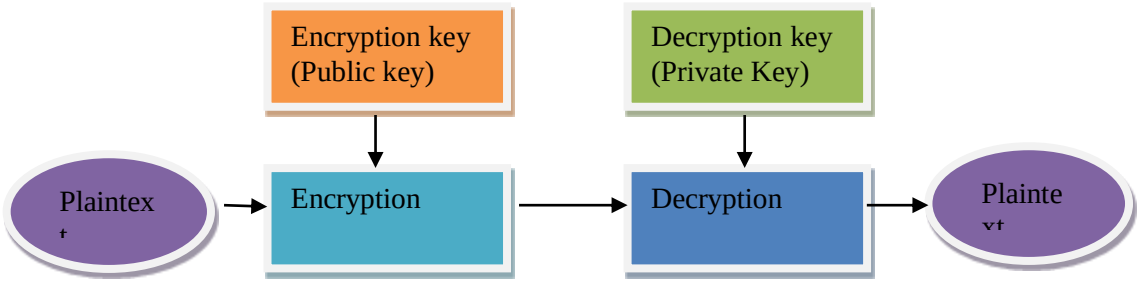


Figure 2.3: Asymmetric Cryptosystem (Ayushi. 2010).

2.3.2 Symmetric Cipher Model

Symmetric Cipher also called private key and it also has known the single key is cryptography in which six elements in all; see figure 2.4: plaintext, encryption algorithm, private and private key, Ciphertexts and decryption algorithm. (Singh et al 2011).This was the main sort of encryption freely known until June 1976(Al-Vahed & Sahhavi., 2011).

Symmetric ciphers use two types of ciphers:

- Block ciphers: is a symmetric key cipher, usually, it encrypts a block of plaintext data (64 or 128 bits).
- Stream ciphers: is a symmetric key cipher, usually, it encrypts one bit or one byte of plaintext data to one bit or one-byte cipher text(1 or 8 bits).

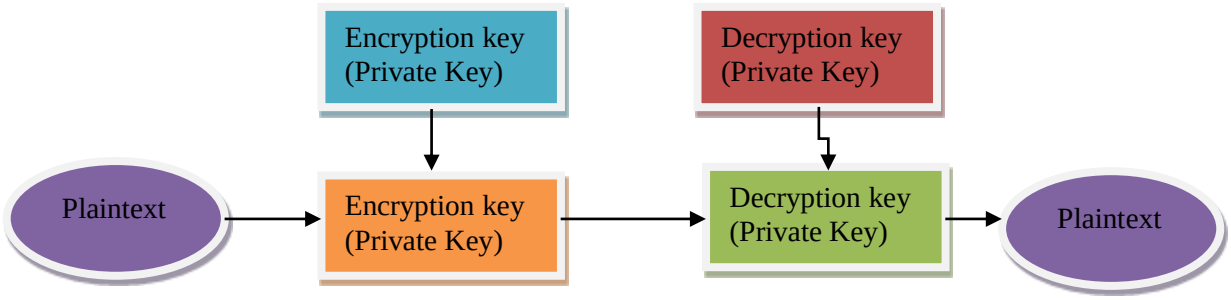


Figure 2.4: symmetric Cryptosystem (Ayushi, 2010).

Sender and receiver must have a knowing of the secret key. A secret key can be initialized by someone or switch between sender and receiver. If you know a person you are exchanging data, you can give them the key in early time, for example, Data protection needed many encryption algorithms, some of these algorithms: Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) (Sangwan. 2012).

2.3.2.1The DES algorithm

The Data Encryption Standard (DES) widely used in the encryption algorithm. DES was developed by International Business Machines (IBM) 1970 by researchers and based on an earlier design by horst Feistel (Biham et al., 1991). Thus, The DES algorithm is foundation stone block for providing data security.

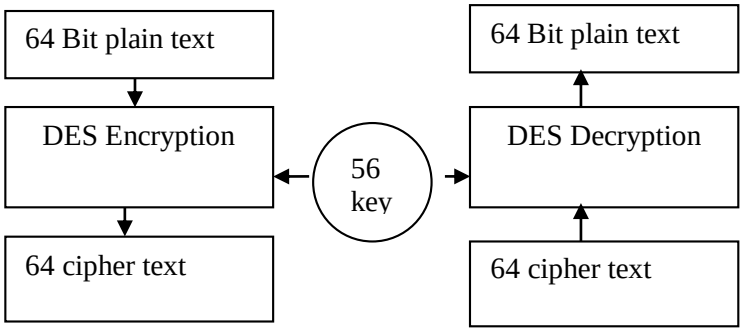


Figure 2.5: DES algorithm for encrypting/ decrypting data ((Kumar et al., 2012).

In details, Kumar. et.al (2012) described the DES. DES is a series of event that occurs during encryption. However, it leads an initial changing on the entire 64 bit of data and 56 bit key. The plain text is subject to an initial permeation to shift the round key and data bit. 64 bit split 32 bit (lift and right) which there are go into around (see the figure).Thus, each Li and Ri going to do an operation. R1 go to function then the result goes to XOR operation but L1 directly taken result XOR. The end halves should be swapped (Kumaret al., 2012)& (Singh et al., 2013).

Paar&Pelzl explained relationship between halving input goes to the next round are:

$$Li= Ri-1$$

$$Ri= Li-1 \text{ xor} F(Ri-1, Ki)$$

The Feistel functions that DES contains many separate stages such as:

- Expansion: here the 32 bit increase diffusion to give 48 bit.
- XOR with round need key. The key 46 split into two 28 such as name Ci and Di. Any round in used left cycle shift.

2.3.2.2Triple DES algorithm (3DES):

The DES algorithm wide uses because it is achieved secures and fast. Thus, cannot break DES but can retrieve key used in encryption by a comprehensive search of 255 steps in the middle. For this reasons, needed powerful than DES in order to protect data.

To address gaps in DES was developed 3DES stronger than DES by expands the key size; application the algorithm three consecutive with different three keys have an effective key length is 168 bits. Most 3DES implementations use two security keys. The result 3DES ciphertext is much harder to break because the total length of the two key 112 bit (Kumar et al., 2012).

- **The encryption is:**

  Cipher text =EK3 (DK2 (EK1 (plaintext))).

- **The decryption algorithm is:**

  Plaintext =DK1 (EK2 (DK3 (cipher text))) (Singh et al., 2013).

Finally, (Alanaz et al., 2010) discussed some factors of DES/ 3DES in its published article as shown in Table 2.2 It advantages triple DES is used block cipher rely your need and takes three likes CPU power(Alanazet al., 2010).

Table 2.2Comparison between DES and 3DES

Factors	Des	3DES
Key length	56 bits	164bits (k1,k2and k3) 112bits (k1 and k2)
Block size	64 bits	64 bits
Developed	1977	1978
Round	16	48

2.4Property of Encryption:

With the development of users' access to the internet and access, large amounts of data found encryption to protect sensitive data so that it became difficult to easily query and searching over encrypted data. Typically example: data storage, webmail, and advertising services. Consequently, encryption scheme found to maintain the digital data that allow for comparison called order-preserving encryption scheme (Popa., 2013) and aggregate queries need homomorphic encryption algorithms (Martinez et al., 2013).

2.4.1 Homomorphic:

(Martinez et al., 2013), Homomorphic encryption has large benefits in cloud computing, especially for those that wish to host encrypted data on cloud providers' servers. The concept is that mathematical operations can be implemented on encrypted data without knowing the private key.

The homomorphic approach can be divided into two types fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE).Full homomorphic Encryptionsupports all arithmetic operations cipher text without decrypt (addition and multiplication). Partial homomorphic encryption supports limited arithmetic operations such as RSA.

2.4.2 Order-preserving encryption scheme:

To deal with range queries on encrypted databases, an order-preserving encryption scheme has been proposed by (Agrawal et al., 2004) OPE is an important method to solve search problem and support all logical operations.

These are a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts and use in in-network aggregation on encrypted data. The issue of OPE it does not allow efficient range queries and also there is some leak of

data. Hence, low security but allows indexing and query processing to be done precisely for unencrypted data (Boldyreva et al. 2009). OPE has been used in indexing and hashing where preserve ordering for data in a database.

The major operation of the database will need the indexing, for example, preserve the order and Binary search is important for indexing. This section described the indexing and order preserving indexing encryption technique to preserve the order data:

2.4.2.1Indexing

Databases spend a lot of their time to finding data and so found data needs as fast as possible. The decisive element of current database management systems is the index. (Cioloca&Georgesc., 2011) defined an Index is a duplicate of the information in the table, sorted in a certain logical manner like (a dictionary and a book index). In other words, Indexes are "construct representation published items in a form suitable for inclusion in some type of database"(Hodge, 1994).

In (Liu& Wang. 2012), an indexing component for range queries is discussed. This component is not strictly order preserving since two different values may be mapped into the same bucket, which is utilized when checking question conditions.

Cloud index presented by Wu et al., (2010).It is designed for online to queries and it offers numerous execution procedures with peer to peer computing. The nature index on a document speeds up choices on the search key fields for the index. In a real world, users tend to query data with more than one key. The search key is not the same as key. The primary key stored in key value store. Secondarily key stored in cloud index.

2.4.2.2 Indexing techniques

The index consists of many clusters such as SQL Server found in each database. All databases employ some horizontal partitioning scheme to store large data tables that support only one clustered index (Cioloca & Georgescu, 2011).

At present, a database index can be sorted out in a B-tree structure called an index shard. The idea is to the division the data into a number of small pieces. This construction starts with a root node (includes a number of index row, key value, and pointer) found on the top level which falls the starting of the index. It is the first data accessed when a data search happens (Wu et al., 2010).

2.5 Hashing (general hash)

2.5.1 Cryptographic Hash Functions

In this section studies cryptographic hash function, Hash functions producing a digest of a message through insert plain text (arbitrary length), convert it to block (fixed size) called a hash value, a message digest, a checksum, or a digital _fingerprint (Tiwari& Asawa., 2010) as shown in figure 2.6.

Northcutt. (2008) explained hashing works by passing data, of any size, through the algorithm which a fixed size output. Cryptographic hashes progress a method for encrypting data; cannot anyone has known the input using only the hashed output. This process can be described as h=H (M) Where h is the hash created by the hash algorithm H and M are the input message.

In their book, (Paar&Pelzl., 2010) the hash functions needed many requirements and very super characterize:

- Free input length and is not constant.
- Fixed and short output.
- Efficient.

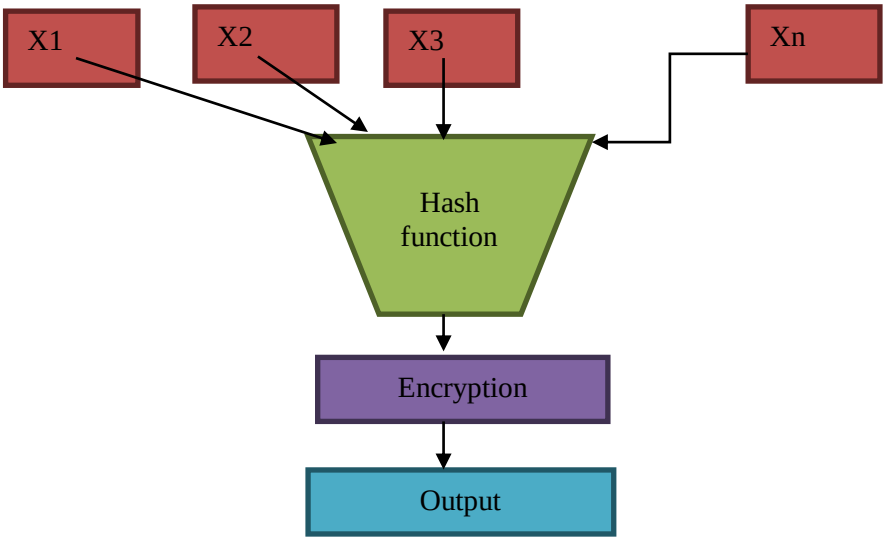


Figure 2.6: cryptographic hash function (Paar&Pelzl. 2010).

2.5.2 Hashing as Encryption Techniques:

Stevens, S., (2007) defined the hashing, in cryptography, (one-way operations) is a system for taking a value then it goes through the algorithm to get a group of information a more compacted structure called a message digest. The operation is not being invertible. All the hash values produced by a given hash have the same size regardless of what the size of the data worth is.

On the other hand, Encryption can be thought of as two-way operations which convert a plaintext into a ciphertext and cipher text into plaintext called decryption (Singh &Gilhotra, 2011).In two operations depend on a key.

2.5.3 Security Requirements of a Hash Function:

Tiwari. et.al (2013) has collected some properties in cryptographic hash function including Preimage resistance, Second preimage resistance and Collision resistance are must use in practical applications of a hash function to message authentication and digital signatures.

- **Considering Preimage resistance** that is called hashes to that output, if an attacker knowing output (h)of the hash function it must be to find data (m) for which h(m)=H.

- **Second preimages resistance**that are, if an attacker is given a data such as m0, it should be impossible for the attacker to change the data and computationally infeasible to find data such as m1 with m1but h(m0)=h(m1).
- **Collision resistance** that is every data to beinga unique hash, the attacker cannot find two data with the same hash. The number input bigger than output (Raj et al., 2013).

2.5.4 Application of hash function in cryptography

Hash functions most imperative device or tool in cryptography, achieving numerous security and objectives such as digital signatures, data integrity and authentication (Shi, Ma, Cote, and Wang. 2012**).**

- **Data signatures or Electronic Signatures:** Historically, digital signatures were the first application of cryptographically secure hash functions. Rabin.(1978) introduced the idea of signing the hash of Cryptographic hash function. In the past, plain text signing a large message directly with a public key cryptosystem.

In this book (Paar&Pelzl. 2010), data signatures receiver contains key public, private signatures and s= sig(x) used to compute data signature. But, we can see very big problem the x is fixed length example |x|< 26 Ɛ B. Hence, must find solution by compress the message x in signing function such as RSA (SΞX^{d}mod n).

The process now works as follows:

- User input data into data signature.
- Secure hash.
- User uses the private key in the input; output it is data signature.
- Mixed Data, data signature and user in new data signature so the other user is sure to the signature all the data.

- **Data integrity:** data integrity is can compute hash the received data, and compare it with a hash of original data, then send through secure channel. If they are the same, there is high secured and achieve confidently. Nowadays, can use MD5, SHA-1, and SHA256. When data has been hashed, changing any of the data will bring about a totally diverse hash.
- **Authentication:** Hash algorithms can be used for the authenticity of information. It is a necessity in computer systems and networks. In this case, two parties communicating over an insecure channel require same a method.

2.5.5 Methods of attack on Hash Functions

In this section will review different types of attacks on hash functions. Attacks on Hash functions which can be categorized into two main categories -Brute Force Attacks and Cryptanalytic Attack. In figure 2.7, (Sobti & Geetha, 2012) clarified the classification of attacks on Hash Functions.

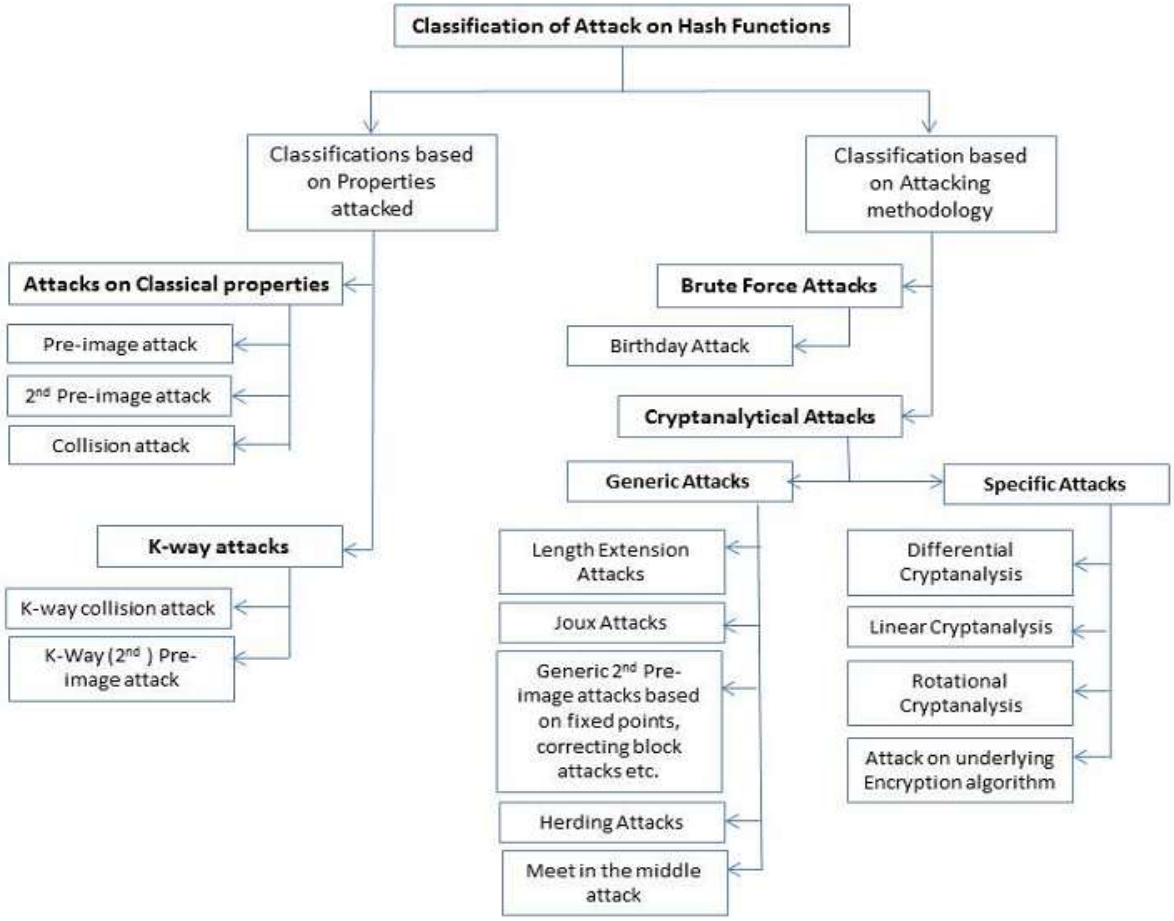


Figure 2.7: classification of attacks on Hash Functions (Sobti & Geetha., 2012).

- **Brute force attack:** Brute-force attack is the most basic attack on cipher that used to try randomly computed hashes to obtain a specific hash digest. Hence, brute force is a trial and error method to obtain a required hash function. As an example of a brute-force attack is a dictionary attack. Spoiling brute force attack, need key space of a cipher should be adequately great (AlAhmad and Alshaikhli., 2013).
- **Cryptanalytical Attacks:** Try to appropriate on hash function such as a preimage attack, second preimage attack, and collision attack. Collisions in hash functions are much easier to find than preimages or second preimages. This methodology breaks on the hash function.

(AlAhmad & Alshaikhli., 2013), As Figure 2.11 shows that crypt analytical attacks on hash functions are classified into two categories:

- **Generic attack:** technical studies used to attack general hash function constructions such as the attack on, SHA-256.
- **Specific attack:** technical studies used to attack specific hash function constructions (based on the hash function itself) such as the attack on SHA-1 and MD5.

2.5.6 Hashing algorithms:

The Merkle-Damgård construction used in designing hash functions such as MD5, SHA-1, and SHA-2.There are many cryptography hash functions used to protect the data; the hash length ought to be sufficiently extensive to prevent an assault from find two or more messages that produce the same hash. Some of these algorithms: message-digest algorithm (MD5), secure hash (SHA-1) and secure hash (SHA-256).

2.5.6.1 Md5 algorithm:

MD5 hash functions are mainly used in message integrity check and password shadow. All people have known MD5 is the most widely used cryptographic hash functions. Nowadays, a hash function designed by Ron Rivest as an enhanced version of MD4in 1992.

The algorithm is presented the md5 by Gupta and Kumar (2014) as follows:

- It takes a variable length plaintext as an input and outputs a 128-bit hash value MD5.
- Hashing plaintext input before doing processes, which divided plaintext to many stages such as p is appended padded bit the length of p with 64 bits; appended padding mean a'1' following by many of '0' to(448 mod 512).
- Padded plaintext divided into parts of 512 bit. Therefore, each piece plaintext

divided into sixteen 32-bit words in the form of chaining variables are presented (A, B, C and D) Then use MD5 compression function. The compression algorithm has four rounds, and each round has 16 operations each round consist steps 1-16, steps 17-32, steps 33-48 and steps 49-64.

Qi = qi−1 + (qi−4 + fi (qi−1, qi−2, qi−3) + wi + ki) ≪si (Gupta and Kumar .2014).

Where:

The (<<<s) item means a binary left shift by s bit. The Plaintext expanding means plaintext word wi is one of (p0….P15), (Gupta &Kumar. 2014).

- In the finish, all of the 64 steps are computed by four functions. The processing block P is applied to four buffer (A, B, C and D), by using plain text Rivest. 1992) and consistent key.
- The four type of function takes input 32 bit and produce a same bit of output (Rivest. 1992).

2.5.6.2 SHA-1

SHA-1 is cryptography hash functions are mainly used to provide data integrity and digital signature. SHA-1 was developed by NIST as US federal processing standard.The algorithm supports plain text any length less than 2^{64}bit as input.

The algorithm is distributed by Gupta &Kumar. (2014) as follows:

- Plaintext as an input and outputs a 160-bit. The SHA-1 would be divided big plaintext into 512 bit. After the plain text has been padding, it must be doing before executing the algorithm.
- Then use SHA-1 compression function. The compression algorithm has four stages, and each round has four operations each round consist steps 0-19, steps 20-39, steps 40-59 and steps 60-79.

- Before hash execution for a secure hash algorithm, SHA-1 initial hash value consists eight 32 bit words in hex. Each round has 5*32 a bit as input (A, B, C, D and E) and input Wj.
- In finish, all of the 80 steps are computed by four round functions. The operation within round in stages is given by:

A, B, C, D= (E+F_t (B, C, D) + (A) <<<5+ W_J+K_i). A. (B) <<<30, C, D.

$$K_i: roundconstant\ k1, k2 \dots k4.$$

$$F_t: functionf1 \dots f4.$$

Tabe2.4: constant key and function used in SHA-1 (Paar&Pelzl. 2010).

Stage	Round	Constant key	Function
1	0-19	K1=5A827999	F(B,C,D)=(B^C)ᵛ (B^D)
2	20-39	K2=6ED9EBA1	F(B,C,D)=B xor C xor D
3	40-59	K3=8F1BBCDC	F(B,C,D)=(B^C)ᵛ (B^D)ᵛ (C^D)
4	60-79	K4=CA62CID6	F(B,C,D)=B xor C xor D

2.5.6.3 SHA-256

SHA-256is cryptography hash functions are mainly used to provide data integrity and digital signature. SHA-256 was developed by NIST as US federal processing standard. It converts an input message into the 256 bits message digest. Hence, must be input less than 264 bits and must be operated by 512 bits in groups.

The following steps describe the algorithm by (Mankar & Nipanikar., 2013):

- Message Padding: input message length =448 mod 512 by padding 1 then 0 into
- Parsing: after padding massage doing parsing message; these message blocks are passed individually to the message expander.
- Message Expansion: the 512 block divided into 32 bit, which is then expanded into 64 words
- Message Compression: after expanding word then go SHA function then initialized hash value H0 (0) – H7 (0).
- Execute algorithm in 46 round.
- Measure intermediate hash value.

There are many differences between the types of hashing functions, but encryption hashing and hashing over encrypted data are important issue types of hashing functions.

2.6 Indexing and Hashing:

The indexing algorithm is not necessary to be applied when a database is securely encrypted and cannot be decrypted. For instance, it is impossible to order the cipher text according to their plain text values. Then given the encryptions, the attacker determine which cipher text coincides, for example, plaintext by simply checking if $E_K(X) < E_K(Y)$. The rules generate B-trees (described at section 2.4.2.2). In contrast, hashing will be feasible even through database is securely encrypted.

The identical plaintext values could be encrypted into different cipher text and can have different hashes and hash bucket. This problem can be addressed by calculating the hash of value before the encryption value. But, Attacker can deduce some knowledge

about the plaintext by simply checking. This problem can be addressed by calculating the hash of value before the encryption value and merge result of hash with plaintexts (Evidokimov & Gunther. 2007).

2.7 Related Studies

This research will classify the literature into parts:

2.7.1 Cloud computing:

Begumetal. **in Data confidentiality scalability and Accountability (DCSA) in cloud computing** have presented an introduction to cloud computing, and which has eight key characteristic which grants it some benefits and these characteristics the following: 1-self healing: a property of healing consist of any application or service running in environment cloud computing. in applications many copies; each copy updating itself without the smallest change in running in cloud computing environment, so that in time fizzle out keep at least one of copies.2-multi-tenancy: the system allows user or owner to exchange infrastructure without briefed of the exchanging. Hence, in which resources are exchanging on a network, host. This means assumed dedicated resources dedicated to a single customer. This is providing security and is not similar. 3-linearly scalable: the system capable smashing burdens work into pieces, the cloud computing linearly scalable.4-service-orinted: the systems are kind that they are created out of other distinguished services or combined distinguished services. 5-Reduced cost: it is important benefits because is not purchased infrastructure and low maintenance. 6-Increased storage: the cloud computing can scale dynamically. 7- Virtualized: The cloud computing environment is a fully virtualized environment; all services and application in cloud computing environment.8-flexible: This is an extremely important characteristic because adapt to changing working conditions (Begum, R.et al., 2012).

Singh et al. in **Cloud Computing Attacks: A Discussion with solutions** have discussed different types of attacks on cloud computing and their respective solutions such as Denial-of-Service attack(DoS), Malware-Injection Attack, Side Channel attack, Authentication attack and Man-In-The-Middle Cryptographic Attacks. Hence, introduced cloud computing and type of attack (Singh et al. 2014).

Patwal&Mittal in **A Survey of Cryptographic based Security Algorithms for Cloud Computing have discussed the Fundamental Characteristics of Cloud Computing** discuss all the Service Models of CC and deployment models of CC, security issues in CC based on the cloud security architecture. Then Advantages, Disadvantages of CC, Security Advantages/ Disadvantages in Cloud Environment finally summarize the some of the Existing Algorithms in Cloud Security (Patwal & Mittal. 2014).

2.7.2 Security in Cloud Computing

Sanjana & Sharma et al. **in Security in Cloud Computing** have described the briefly details of cloud computing and type of services and security issues and some challenges for data security in cloud environment and investigate the several approaches for security in cloud computing and finally provide a reliable security in a cloud computing for future work (Sanjana & Sharma et al. 2012).

2.7.3 Cryptography

Al-Vahed & Sahhavi described a short look at modern approaches in cryptography. So, terminology, history, and types of modern cryptography will be talked (Al-Vahed & Sahhavi. 2011).

In Bouganim &Guo **Database Encryption** have interested in a security, model for a database that prevents an intruder to filtering on the network and taken information about

the database. Thus, that means database encryption the purpose of database opacity by keeping the information hidden to any unauthorized persons. Encryption can provide strong security for database but developing a database encryption strategy must take into consideration the following issues:

- Where the encryption should be performed?
- How much the encryption technique ensures security?
- How the encryption technique works?
- Who know about the encryption key?
- How to minimize the impact of database encryption on performance? (Bouganim&Guo. 2011).

Ayushi in A **Symmetric Key Cryptographic Algorithm** has explained cryptography definition and then identified two basic types of cryptography: Symmetric Key and Asymmetric Key. Also, described cryptography, various symmetric key algorithms in detail and then proposes a new symmetric key algorithm. Algorithms for both encryption and decryption are provided. The advantages of this new algorithm over the others are also explained. At conclusion Ayushi explained important aims like Confidentiality, Data integrity, Authentication etc. and important algorithm on reducing cost (Ayushi, 2010).

Tebaa & Elhajiiin Secure **cloud computing through homomorphic encryption** have presented cloud computing definition, and then identified the different structures of cloud and services models as public, private, hybrid, and community. Also, they defined several cloud service as Software as a Service, Platform as a service and infrastructure as service. They proposed approach concept of homomorphic encryption and division homomorphic FHE, PHE (Tebaa & Elhajii, 2013).

Chou in **Security threats on cloud computing vulnerabilities** have explained cloud computing definition and then identified cloud service models: infrastructure as services, platform as services and software as services. Also, described the taxonomy of cloud security threat in detail and countermeasures. At conclusion Cloud computing is in continual development in order to make different levels of on-demand services available to customers. While people enjoy benefits cloud computing brings, security in clouds is a key challenge and in this paper, Chou examined the security vulnerabilities in clouds from three perspectives (abuse use of cloud computational resources, data breaches, and cloud security attacks), included related real world exploits, and introduced countermeasures to those security breaches (Chou. 2013).

Debnath et al presented comparative study between of different encryption algorithms kind; AES, DES, and 3DES. Then presented into nine factors. At the conclusion, he found from the experimental results, proved the 3DES is better than DES discussed in this paper (Debnath et al., 2014).

Kummar et al presented the comparison between DES and RSA algorithm. There are two main features that are specified and differentiate one algorithm from another are the ability to secure and protect the data against attack and speed of encryption and decryption. Also, described the performance of three most useful algorithms: DES, 3DES, and RSA. At a conclusion DES and RSA are discussed with their mechanism and explained DES is the secret key. But RSA large amount of time to perform encryption and decryption operation simulation result showed that des better than RSA (Kummar, jakhar & Makkar).

Sarode, Giri & Chopde in **The Effective and Efficient Security Services for Cloud Computing** have proposed the Cloud systems such as encryption/decryption system, storage system, and the user interface implementation system. Thus, needed understand the system of encrypt and decrypt. The system of encrypt and decrypt using RSA algorithm because that achieve credibility, which running under the public key and private key (Sarode, Giri&Chopde. 2011).

Karthik& Muruganandam in **data encryption and decryption by using triple DES and performance analysis of cryptosystem** have presented a technique for secret communication using cryptography. Also, presented an introduction about DES, and which has history DES, cryptography. In cryptography explained five goals. 1-authentication: the sender and receiver must be verified. 2-confidently: it mean only the authenticated people are able to interpret the message or content and no one else. 3-Integrity. 4-Non-repudiation. 5- Service Reliability and Availability. Hence, explained advantages, disadvantages and objective then motivation. Then Karthik explained performance comparison between the most encryption algorithm DES, 3DES, and AES. Through an overview of encryption and decryption explained type encryption contact manual encryption, transparent encryption, symmetric encryption and asymmetric encryption. Also, explained type decryption contact symmetric, asymmetric and hashing (Karthik & Muruganandam., 2014).

Roshdy et al proposed hash algorithm has been designed to satisfy the different level of enhanced security and to resist the advanced hash attacks by increasing the complexity degree of proposed hash algorithm (MD5 and SHA-256) (Roshdy, fouad and Dahab. 2014).

2.7.4 Performance

Sehgal & Narwal **in Analysis of Performance for Multi-Tenant Application through Cloud SIM** have explained the algorithms to keep both performance and data secure but flexible enough to allow for expandability. This description allows providing solve problem data for security and performance because start adding new hardware/update existing hardware in a web cloud. Hence, Sehgal &Narwal executed an application on cloud simulation. There are introduced about cloud computing and models used in cloud computing (Sehgal&Narwal. 2015).

A study in (Abdul.Elminaam. et al) is presented a performance evaluation of selected symmetric encryption algorithms such as AES, DES, 3DES, RC6, Blowfish and RC2. They were implemented, and their performance was computed by several steps by simulation. Also, we find that 3DES still has low performance compared to algorithm DES (Abdul. Elminaam, Abdul-kader & Hadhoud. 2008).

2.8 Summary:

The literature review helped us a better insight with reference to cloud computing, different models, and deployment of cloud computing, Cloud computing attack, a current security issue. During the reviews, it is noted that many of research is going on in cloud computing security issues and how to overcome the security issues and to gain cloud users confidence. Understanding different encryption algorithms like DES and 3DES. Understanding different Hashing algorithms.

CHAPTERTHREE

The proposed solution

A Combination of Encryption Hashing and Hashing

Over Encrypted Data

3.1 Overview:

This chapter explains the most important concept of the research thesis it presents. The most associated studies in environment tool such as hardware and software and proposed the model.

3.2 Introduction:

Recently, cloud computing is considered the most important issues in the field of cloud computing, which got a great importance by the researchers. Cloud computing for a number of other a advantages such as low cost and ease characterized by access to data anytime and anywhere, the same time is the most important flaws appeared lack of security. The data security issues that have been put forward, where they are treated by means of encryption. Encryption lead to new problems, including the failure to preserving to functions mathematical, or specify the location of data in a database, but found techniques to maintain the order within the database encryption, a technique order preserving data. The search for data by indexing, as well as the Hash Techniques was used gain access to data in a prompt manner and less cost way of indexing and using hashing encrypted data. But this solution impact on a performance level, access to data and reduces security. Several studies have proven there is a problem because of the collision contains several texts on the same hash value. It has been solved the previous problem dependent on the (Evidokimov & Gunther. 2007), they used the hash algorithms before encryption. Also, but this solution impact on a performance level, access to data and reduces security. It has been solved the previous problem dependent on the (Evidokimov & Gunther. 2007), they used result hash with plaintexts then pass to encryption.

The main goal of the research will explain the effect of hashing and encryption process, with several parameters to study their effect on it is performance. These parameters are: encryption algorithm, hashing algorithm, key size, and text size. This research looks deeply into the optimal values of these parameters to find the best performance (CPU speed) and the best profit security.

The following will explain these parameters in detail.

- Data size

  This thesis used a different sizes of a text file where text file changes were relying upon need. That the data sizes utilized as a part of our work contain (10 Kbyte, 100 Kbyte, 500 Kbyte, 1500 Kbyte, 3 Mbyte, 10M byte and 100 Mbyte),Although data type at this thesis was not taken as a parameter.

- Hashing techniques

  This thesis widely used hashing algorithms for their soundness and usability. It used different algorithms such asMD5, SHA-1, and SHA-256. These algorithms have been introduced in chapter two.

- Encryption techniques

  This thesis studied well-known encryption algorithms for their soundness and usability. That used different algorithms such as DES and triple DES. These algorithms have been introduced in chapter two.

- Key size text folder

In order to accomplish the research of this thesis, we decided not take into consideration the key size as one of owner parameters, different type of encryption techniques will not

allow us to generalize the same key each time.

In DES and triple DES, the keys are fixed. Only 56 bits for DES and 112 and 168 for triple DES.

3.3 Environment tools and setting:

In the following we will give several ideas about the hardware and software.

3.3.1 Hardware Used in This Research:

We have conducted tests on computer hardware/ software. We execute our experiments using a SAMSUNG with Processor: Intel® core ™ i53210M CPU @ 2.50GHz (4 CPUs), 4096MB memory.

3.3.2 Software Used in This Research:

The tests are conducted on Windows 8.1 Home pages for C sharp. C sharp software was used to encrypt data on different keys, plain text, encryption algorithm and hashing algorithm.

Windows 8.1 contains C sharp to allow access to encryption and hashing algorithm then run it.

3.4 Experiments phases:

The phases are:

3.4.1 Studying Phase

In this phase, we studied several types of hashing functions and encryption functions with their parameters.

3.4.2 Design and Implementation Phase

This phase has collected texts and keys from random samples, studied and used the encryption functions and hashing algorithms. This dissertation used the two approaches (encryption hashing and hashing over encrypted data) was proposed by (Evidokimov & Gunther, 2007).

This thesis has designed and implemented the hashing techniques and encryption technique with several parameters. Then it runs the experiments with all the parameters the performance (the execution time for both the process and percentage CPU).

The proposed model has consisted of four components:

- Choosing file text then uploading text ((10 Kbyte, 100 Kbyte, 511Kbyte, 1500 Kbyte, 3 Mbyte, 10M byte, and 100 Mbyte).
- Choosing key text then uploading text.
- Running file text and key text with encryption and hashing; where plain text hashed by hashing algorithm such as MD5. Then the execution time will be calculated (DES and Triple DES) with different keys. Later the result will be merged with the plaintext. Then the execution time will be calculated (DES, Triple DES).
- Calculate the performance.
- Compare the performance and between (encryption hashing and hashing over encrypted data) with regard to execution time and CPU percentage.
- Implementation Detail of main algorithms.

3.4.3 Encryption of hashing and hashing over encrypted data

This section illustrates the effect of hash encryption, hashing and encryption with different keys size and with different data input size. This section first explain algorithm for encryption hashing.

Figure 3.1 shows the proposed model methodology. The steps of the methodology are used to conduct a performance analysis for hashing over encrypted data to find the loss of performance by using encryption, hashing functions, and other parameters. A methodology is a combination of descriptive and quantitative research. It was mainly based on studying and implementing encryption, hashing function the performance with several parameters. It has used quantitative research for doing many experiments and analyzing the performance of the system. Where we building the experiments, as following:

- Collecting data (plain text)
- Collecting many hash functions.
- Collecting some encryption techniques (DES, and 3DES).
- Collecting several hashing techniques (MD5, SHA-1, and SHA-256).
- Run many experiments for several (hash function, an encryption technique, and compute duration time and CPU percentage.
- The result.

The idea was to run all the parameters at the same time on the two approach (encryption hashing and hashing over encrypted data) to calculate percentage CPU and execution time for two approaches.

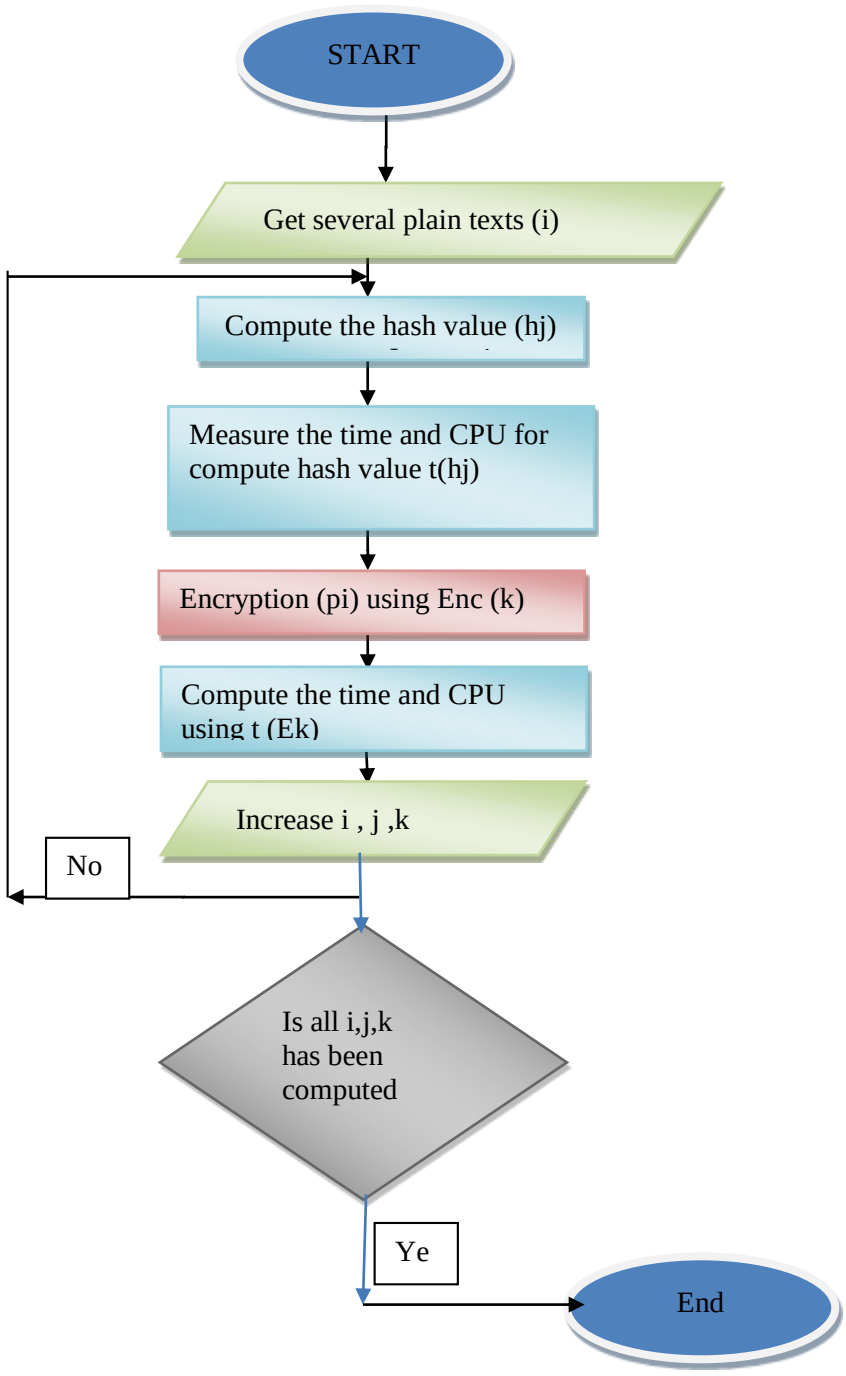


Figure 3.1: The proposal model methodology

Where:

■i: plain text

■j: hashing algorithm

■k: key size used in an encryption algorithm.

The main steps for the proposed solution are summarized in figures 3.2:

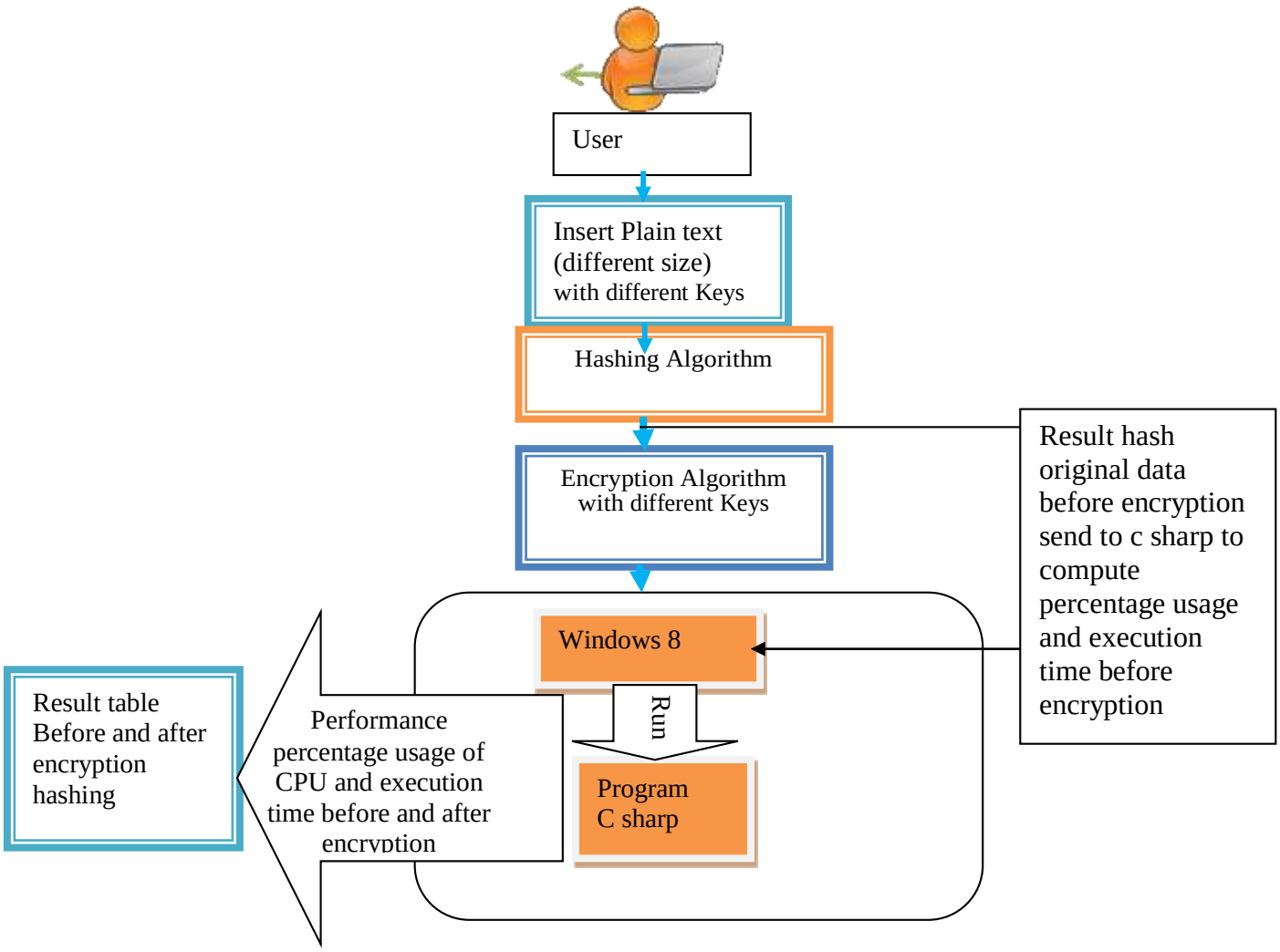


Figure 3.2: Encryption hashing.

Figures 3.2 flowchart work as follow:

- Upload several plaintexts (data is in different size)
- Send it to the hash function (MD5 or SHA-1 or SHA-256).
- Then encrypt the resulting hash by using encryptions algorithm.

This illustrates in figure 3.2 summarize steps for hashing and encrypted the value of that hash.

To see the effect of merging the result of hashing with original data. The following flow chart figures 3.3 illustrate the main steps.

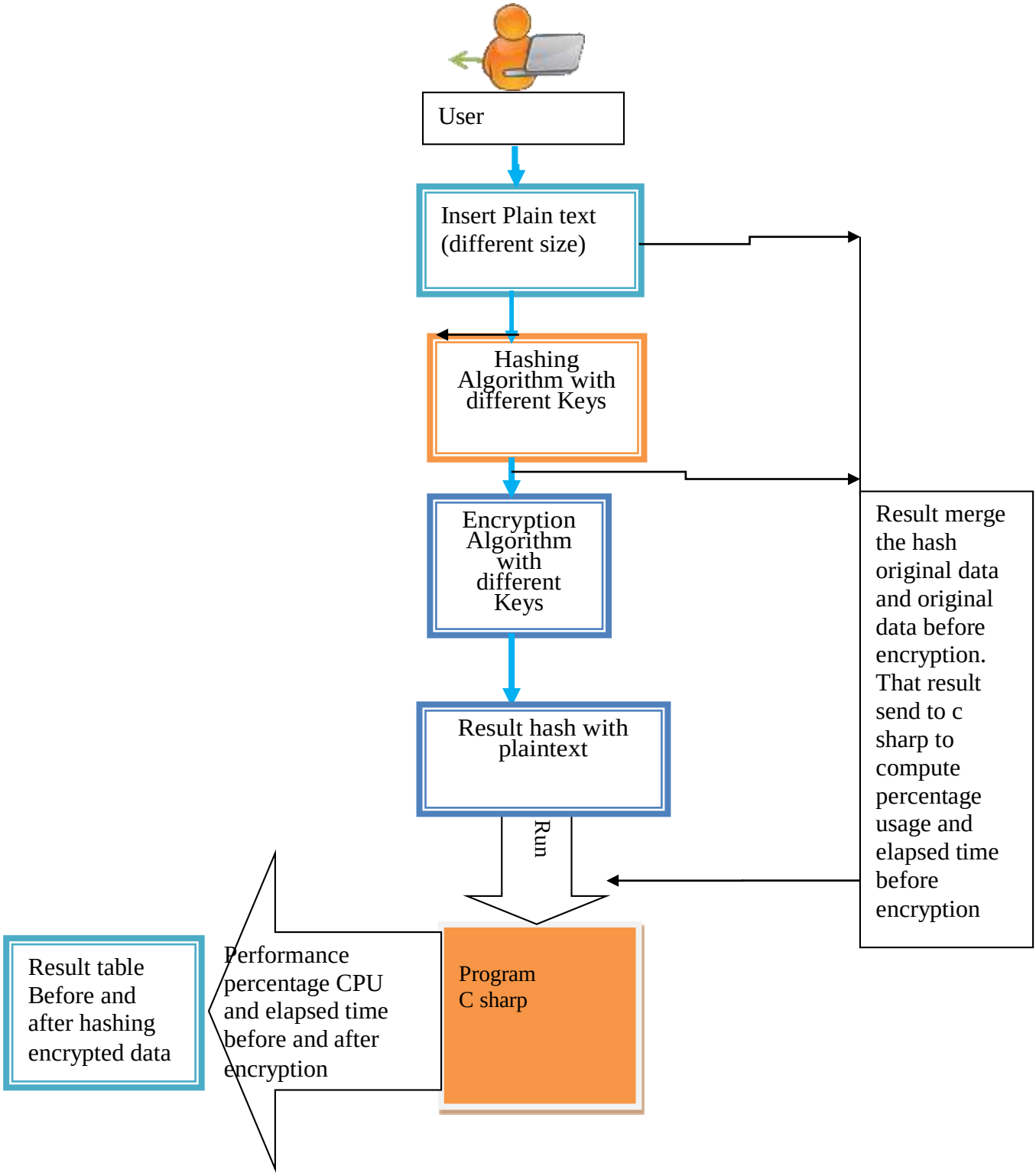


Figure 3.3: hashing over encrypted data.

The previous flow chart works as follow:

- Upload plaintext
- Send it to the hash function (MD5 or SHA-1 or SHA-256).
- Add the result digests to the plaintext (merge).
- Then encrypt the text result merge using encryptions algorithm.

As shown by figures 3.3, the result table has been computed before and after hashing encrypted data, since the input plaintexts are merged with hash techniques (output for hashed plaintexts), then the proposed technique computes the execution time and percentage CPU to find effect all parameters on performance.

3.4.4 Evaluation Phase

In this phase, in order to accurately analyze the results, the following steps were taken:

- Measure the execution time/ CPU percentage before and after encryption hashing
- Measure the execution time/ percentage CPU before and after hashing over encrypted.

The results will be summarized in tables. The main fields are:

Name of original data, Size data, Key name, Name hash, Hash size ,Time execution of before encryption hashing , CPU duration of before encryption hashing, Encryption Name, Time duration of encryption hashing (time), CPU duration of encryption hashing, Merge result hash and original data, Merge Size, Time duration of before encryption

hashing (over encrypted data), CPU execution of before encryption hashing (over encrypted data), Encryption name, Encryption merge size, Time execution hashing over encrypted data and CPU during hashing (over encrypted data).

5 Analysis of the Result

This phase analyzes and finds relationships between encryption hashing and hashing over encrypted data and used different parameters such as encryption algorithms, hashing algorithms, different file size (text) and different keys. We use the average of CPU time to determine optimal a performance, and profit of security. Loss of performance means the difference between the averages (Time) for two different parameters the list and divided this difference on maximum average.

To evaluate the performance, for the send, we needed to answer the following questions from tables:

- What are the average duration time in minute and seconds (m:s) and CPU utilization (percentage %) for each step?
- What is the effect of encryption algorithms on the performance?
- What is the effect of hashing algorithms on the performance?
- What is the effect of the key on the performance?
- What is the effect of the Data size on the performance?

These question will be answered in chapter 4.

CHAPTERFOUR

Results and Analysis

4.1 Overview

This chapter discusses in details the experimental results and their analysis. It has been divided into four sections. Section 4.1 introduces the chapter. Section 4.2 explains implementation on software. Section 4.3 explains the evaluation metrics. Section 4.4 explains the experiments results and analysis.

4.2 Introduction

The aims of this dissertation is to find better algorithms; where encryption and hashing achieved high security and high performance, by comparing among several encryption hashing and hashing over encrypted data (before and after encryption).This research it recorded execution time and percentage CPU for several parameters that effect the performance. The solution has implemented and designed to analyze experiments studying and compare the performance algorithms for several encryption hashing and hashing over encrypted data. Plaintext or data size have been run with several parameters (keys, encryption algorithms, hashing algorithms). In order to accomplish the research of this thesis, we decided not take into consideration the key size as one of owner parameters, a different type of encryption techniques will not allow us to generalize the same key each time. For that the effect of key size was very little and the accuracy of it was not good for that reasons. Process execution time and execution CPU have been recorded before and after for each step(encryption hashing and hashing over encrypted data). This thesis measured the execution time and CPU percentage for each step to

study performance. The thesis used three hashing techniques these are:

- MD5: the md5 algorithm is presented by Gupta and Kumar (2014) and explained in details in chapter 2.
- SHA-1: SHA-1 is cryptography hash functions are mainly used to provide data integrity and digital signature. SHA-1 was developed by NIST as US federal processing standard. The algorithm supports plain text any length less than 2^{64}bit as an input.
- SHA-256: SHA-256 is cryptography hash functions are mainly used to provide data integrity and digital signature. SHA-256 was developed by NIST as US federal processing standard. It is convert an input message into the 256 bits message digest. Hence, must be input less than 264 bits and must be operated by 512 bits in groups.

The result of the experiments were compared depended on the performance of algorithm used on encryption hashing and hashing over encrypted data. Encryption hashing means different size plaintext pass into hash function, which result hash entering encryption. but, hashing over encrypted data mean different size plaintext entering into the hash then result in hash merge with plaintexts; where result in merges hash function passing into encryption algorithms.

4.3 Execution Evaluation Metrics

This chapter has been studied many evaluations metrics to analyze the experiments.

- loss of performance

This metric used to find the loss of performance as execution time and CPU percentage. The loss of performance calculated by a difference between the max value and minimum value then divided it on the maximum value. These metrics computed to determine which parameters provided loss of performance.

$$\text{Loss of performance}=\frac{Max\ value\ of\ both\ technique-Min\ value\ of\ both\ technique}{Max\ value\ of\ both\ technique}*100\%.$$

- determine the best performance

Find best performance algorithms (encryption and hashing) by Average time and average CPU calculated by addition parameters then divided these parameters on number.

4.4 Experiments Results:

We have built our proposal project using visual studio C sharp, where we have implemented three hash functions are (MD5, SHA-1 and SHA256) and three encryption algorithms (DES and Triple DES). Then varying the hashing and encryption between these functions. Hence, we have used encryption purpose.

The programme first upload file with different size then pass it to MD5, take the key value and pass it to one of the encryption algorithms. This process is repeated for all encryptions and hash functions and then we compare results obtained in each case.

- **Scenario 1 (encryption hashing):**

We have obtained digest with size: 32MB for MD5, 80 MB for SHA-1 and 80 MB for SHA-256. And in each test we have calculated the percentage CPU and executiontime before encryption and after encryption. The goal is to find which of the previous encryption algorithms will require more percentage (utilize CPU) and execution time. More time needed for encryption. Three phases have considered summarizing these phase.

Table 4.1: running phase between hashing& encryption.

Phase1	MD5	DES
		Triple DES
Phase2	SHA-1	DES
		Triple DES
Phase 3	SHA-256	DES
		Triple DES

We compare our results in phase1, phase 2 and phase 3 to find the best performance in term average percentage CPU and average execution time. Our results show which combination between a hash function and encryption algorithm is better in term of performance calculating the loss of performance. The following explain this steps in detail.

4.5 MD5

This Algorithm takes different sizes but gives fixed size. As aim output appendix I shows the whole md5 results. For a better understanding of appendix I, results are summarized in several tables and the tables of which is dedicated to calculating the length of time and the percentage CPU before encryption. Let's take Experiments number 1 as an example. This process uses MD5 with file size 10 Kbyte before encryption. The execution time before encryption were 49 (m.s). The CPU of percentage before encryption were 1.190%. Table 4.2 sampled these results.

Table 4.2, sample of MD5 results (CPU).

Experiments numbers	File size Kbyte	Hash name	Hash digest	CPU %	Time(m.s)
1	10 Kbyte	MD5	32	1.190	49
2	100 Kbyte	MD5	32	2.623	11
3	500 Kbyte	MD5	32	1.318	3
4	1500 Kbyte	MD5	32	1.727364	10
5	3 Mbyte	MD5	32	0.4100	20
6	10 Mbyte	MD5	32	1.482	47
7	100 Mbyte	MD5	32	21.519	274
Average			32	42.324	59.143

Figure 4.1: Draw the relation among CPU time and execution time

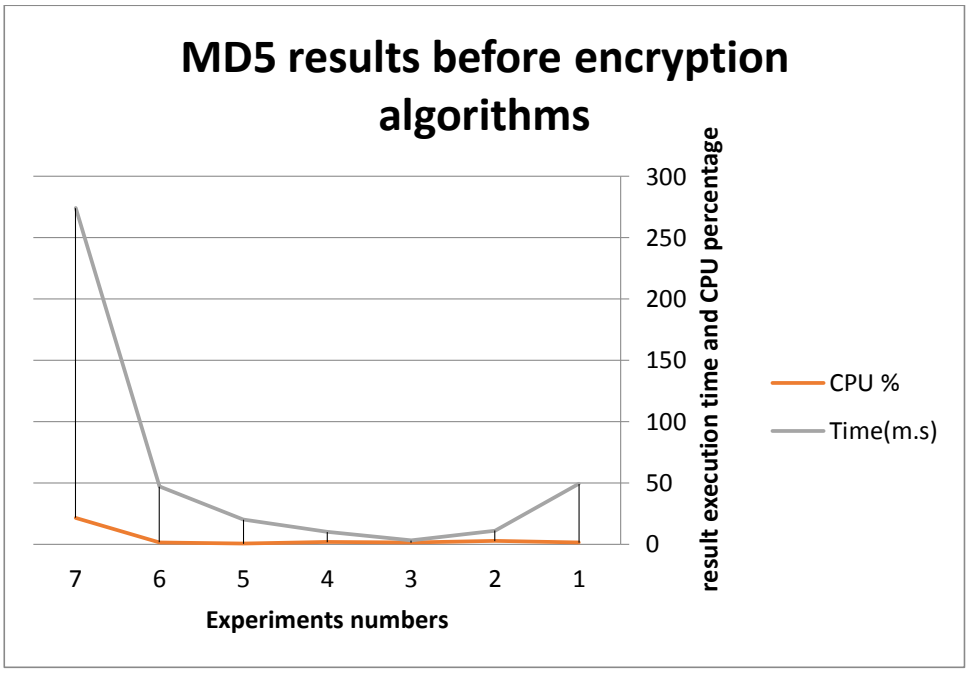


Figure 4.1: MD5 results before encryption algorithms

From table 4.2and figure 4.1, it is clear that CPU percentage increased with file size. And execution time increased with file size. For their result above 500 KB.

The total time for the seven experiments was 414 (m.s) and total CPU for seven experiments 30.27% while the average time hash all experiments equal59.143 (m.s) and the average CPU percentage hash equal 42.3%.

In this line study the result for the key. The key size was one the parameters that we intended to investigate, but we noted that some encryption and hashing techniques did not allow us to choose the size of a key. For that, the effect of key size was very little and the accuracy of it was not good for that reasons. Encrypt the 32 digest with DES and Triple DES. Table 4.3, show the encryption percentage CPU for several experiments has been done. The value of each hash (digest) with DES and Triple DES.

Table 4.3 sample of MD5 results (CPU).

experiment number	**percentage CPU%** **DES**	**percentage CPU %** **Triple DES**
1	2.00	23.33
2	3.56	9.16
3	2.15	10.44
4	2.093	3.08
5	3.064	1.895
6	2.79	1.91
7	49.945	25.699
AVERAGE	9.37	10.79

For clarification the results in the previous table, the following have been drawn in figures 4.2.

Figures 4.2, shows the encryption percentage CPU for encryption algorithms DES and Triple DES with hash MD5.

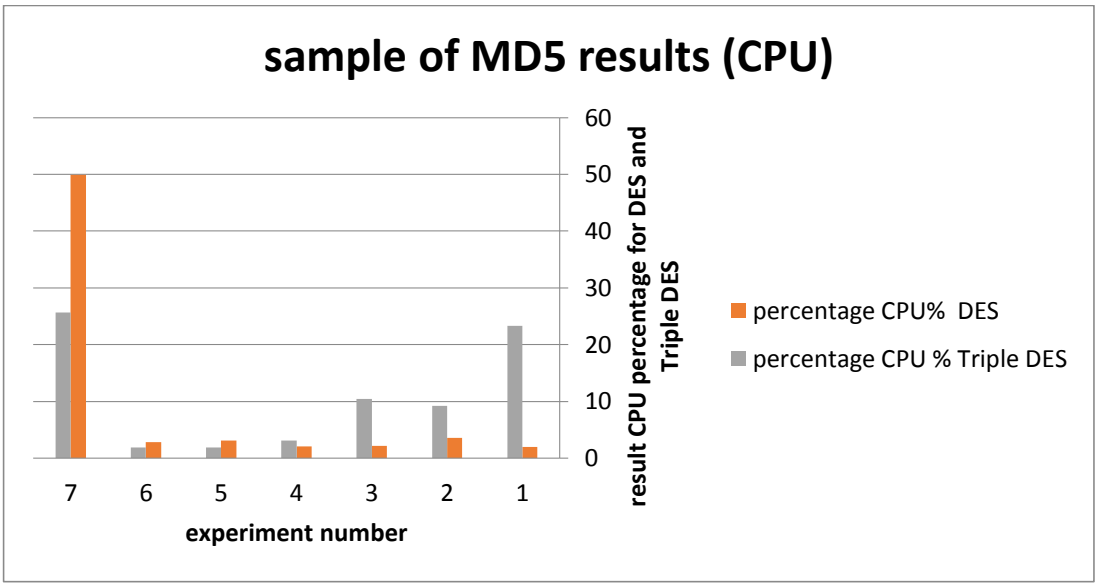


Figure 4.2: encryption percentage CPU

Figure 4.2 shows for the experiments (1, 2,3 and 4) the CPU percentage with DES less than Triple DES. Experiments (5, 6 and 7) CPU percentage for Triple DES is less than DES.

Figure 4.2 shows that the performance of CPU for DES is better than Triple DES. Since the lower CPU percentage is butter than higher CPU percentage. Figure 4.3, shows the average percentage CPU for encryption with hash MD5.

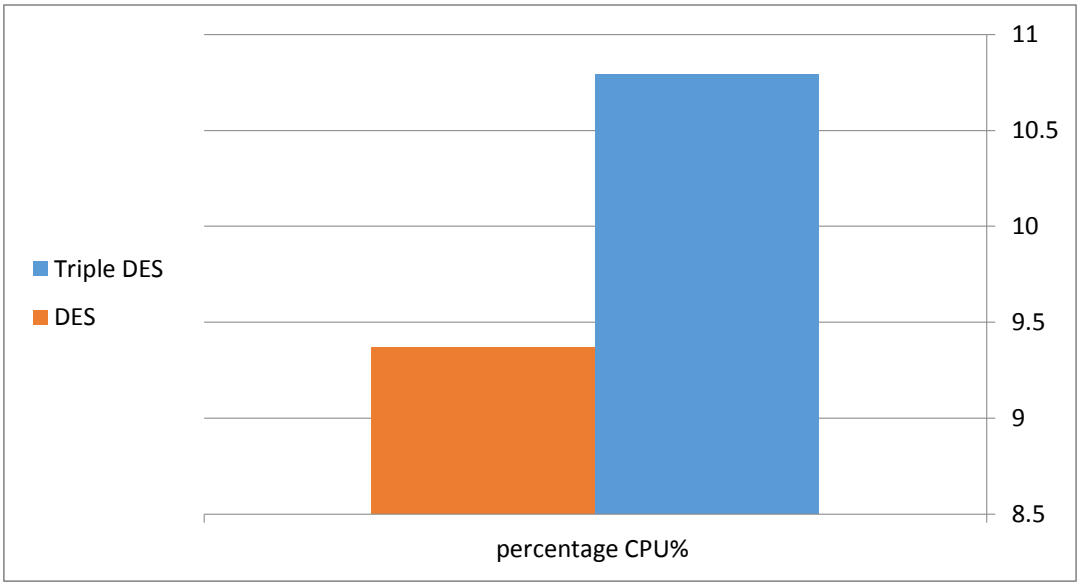


Figure 4.3: average encryption percentage CPU

From figure 4.3, it is clear average encryption for percentage CPU for DES and Triple DES. Figure 4.3 shows that the performance average CPU percentage for DES is **better** than Triple DES.

Table 4.4, show the encryption time for variable file size and using MD5.

Table 4.4, sample of MD5 results (execution time).

experiment number	Execution time (m.s) DES	Execution time (m.s) Triple DES
1	96	54
2	13	13
3	5	4
4	12	12
5	24	22
6	49	49
7	274	276
Average	67.57	61.43

For clarification, the results in the previous table have been drawn Figure 4.4.Figure 4.4, shows the encryption execution time for encryption algorithms DES and Triple DES with hash MD5.

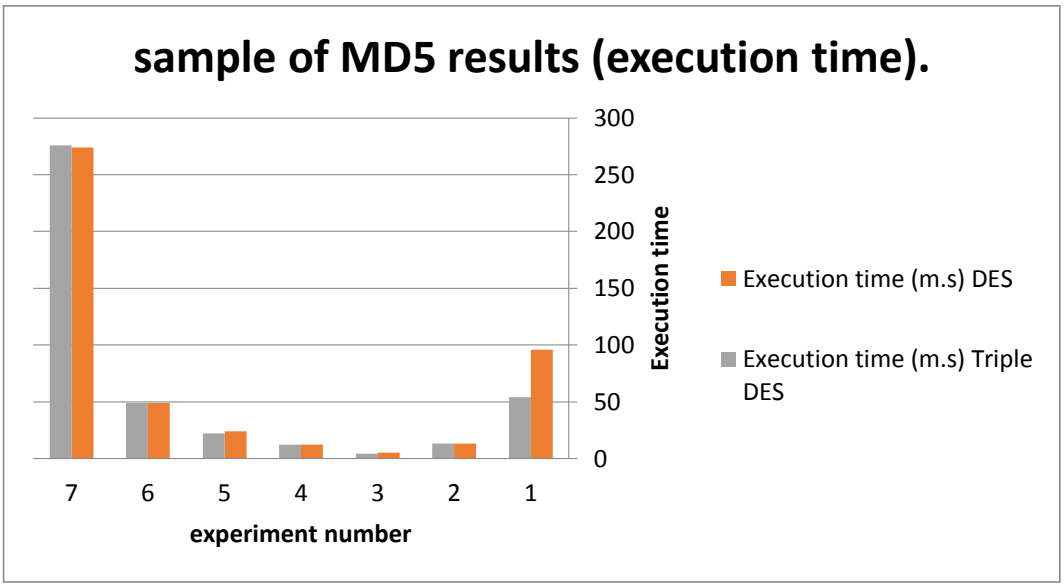


Figure4.4: encryption execution time.

Figure 4.4shows that for the experiments (1) execution time for **Triple DES** is **less** than **DES**. Experiments (3) execution time for **Triple DES** is **less** than **DES**. Experiments (5) execution time for **Triple DES** is **less** than **DES**. Experiments (7) execution time for **DES** is **less** than **Triple DES**.

Figure 4.5, shows the average encryption execution time for encryption algorithms DES and Triple DES with hash MD5.

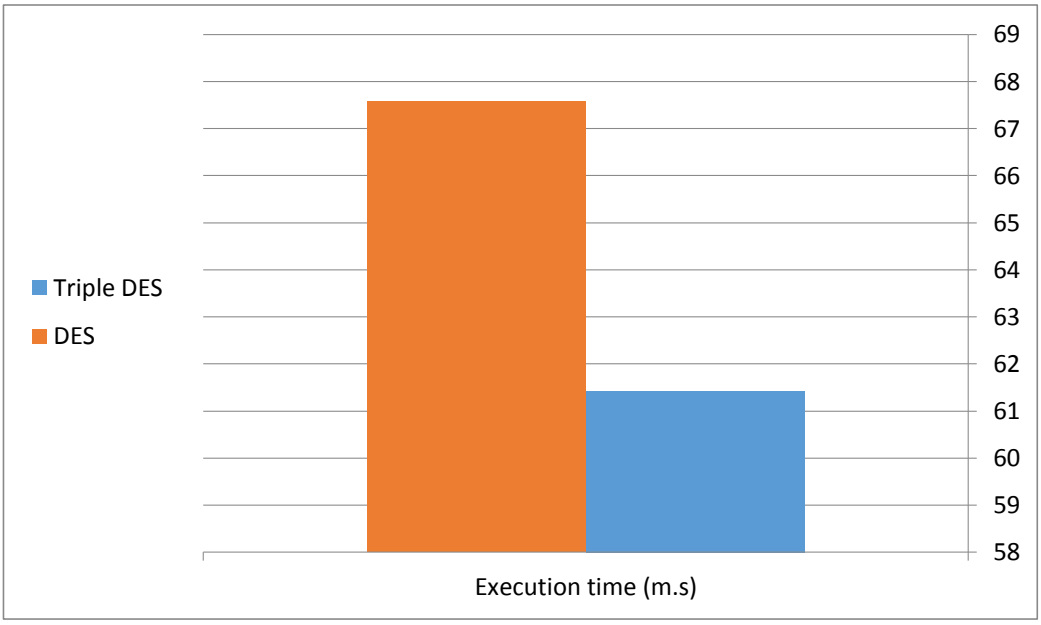


Figure 4.5: average encryption execution time

From figure 4.5, it is clear average encryption execution time for DES and Triple DES. Figure 4.5 shows that the performance of average execution time for **Triple DES** is ***better*** than **DES**. Since the lower execution time is better the higher CPU percentage.

4.5.4 Summarize result MD5 after encryption (encryption hashing):

Table4.5: summarize result MD5 for percentage CPU after encryption with different keys.

Summarize **CPU percentage**	**DES**	**Triple DES**
After encryption with key	9.37	10.79

The previous table, shows hash MD5 after encryption which the key size of the parameter that we intended to investigate. We noted that many of hashing techniques and encryption techniques allow us to choose the size of the key. For that the effect of key size was very little and accuracy of it was not for that reason; DES better than Tripe DES. Where DES with **ALL better** than Triple DES.

Table4.6: summarize result MD5 for execution time after encryption

Summarize execution time	**DES**	**Triple DES**
After encryption	67.57	61.43

The previous table, shows result hash MD5 after encryption which the key size was the parameter that we intended to investigate. We noted that many of hashing techniques and encryption techniques allow us to choose the size of the key. For that the effect of key size was very little and accuracy of it was not for that reason; Triple DES with **all is better** Than DES.

Scenario 2 (hashing over encrypted data):

The programme merge different size of text with result hash MD5, take the key value then pass it to one of the encryption algorithms, take the key value and pass it. This process is repeated for all encryptions and hash functions and then we compare between results obtained in each case.

The system work as follow:

- Upload file plaintext.
- Sent it to hash function (MD5 or SHA-1 or SHA256).
- Add the resulting digest to the plaintext.

- Then encrypt the text using encryption algorithms. And in each test we have calculated the percentage CPU and execution time before encryption and after encryption algorithms will require more percentage (utilize CPU) and execution time. More time needed for encryption. Three phases have considered summarized these phase.

Table 4.7 Running phase between merge hashing with plaintext and encryption.

Phase1	result MD5 merge with plain text	DES
		Triple DES
Phase2	result SHA-1 merge with plain text	DES
		Triple DES
Phase 3	result SHA-256 merge with plain text	DES
		Triple DES

We compare our results in phase1, phase 2 and phase 3 to find the best performance in term average percentage CPU and average execution time. Our results will show which combination between hash function and encryption algorithm is better in term of security calculating the loss of performance. Table 4.7, Show the percentage CPU and hash time for variable file size using merge result MD5 with different plaintext size before encryption such as DES.

As an output appendix II, shows the whole merge result md5 with different data size.

Table 4.8 give a sampled these result hash with different data size before encryption.

Table 4.8, sample merge result hash with different data size.

Experiment number	Hash Name	Result merge size	CPU %	Time(m.s)
1	MD5	10.476 Kbyte	30.31	3,053.00
2	MD5	100.868 Kbyte	0.87	3,021.00
3	MD5	500.298 Kbyte	56.42	3,035.00
4	MD5	1500.016 Kbyte	0.75	3,070.00
5	MD5	3.072000 Mbyte	53.08	3,135.00
6	MD5	10.239154 Mbyte	22.34	3,371.00
7	MD5	100.399904 Mbyte	27.28	6,352.00
Average			27.29286	3,576.71

Figure 4.6, shows draw the relation among CPU time and execution time

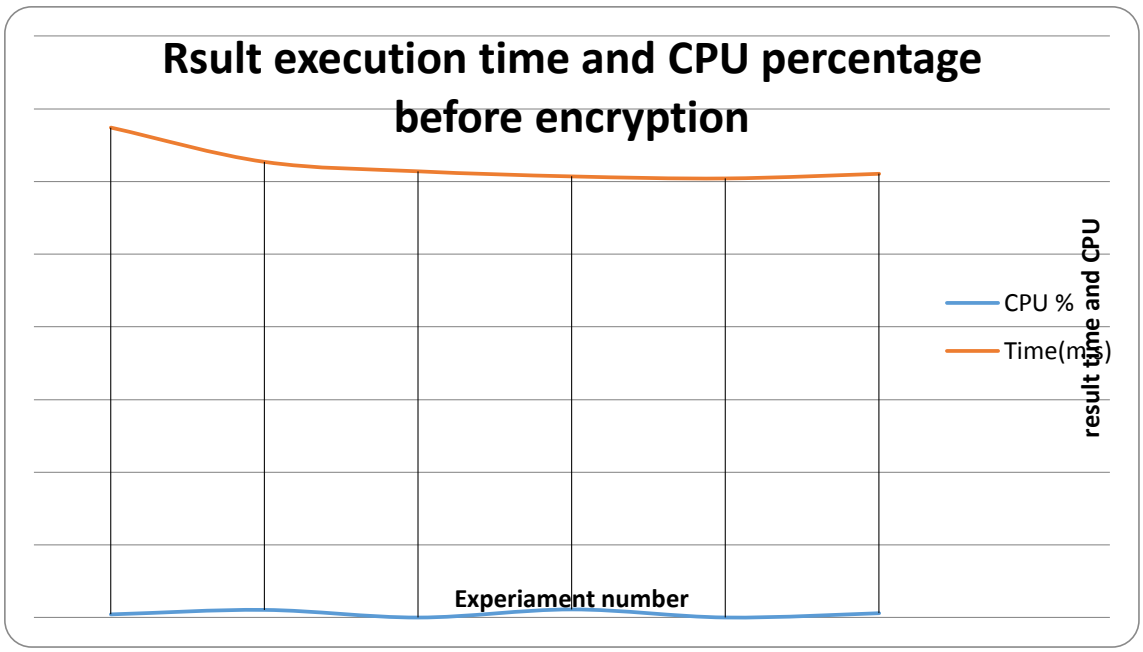


Figure 4.6: result (merge) for time and CPU before encryption.

From table and figure 4.6, it is clear that execution time increased with file size. And CPU increased for the value above 56.

In this line study the result for a key.We have chosen key size was one the parameters that we intended to investigate we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that, the effect of key size was very little and the accuracy of it was not good for that reasons with DES, Triple DES and RSA. Table 4.9, show the encryption percentage CPU for variable file size merge with result MD5 before encryption algorithms.

Table 4.9, result the encryption percentage CPU for variable file size merge with result MD5 after encryption algorithms.

experiment number	**percentage CPU% DES**	**percentage CPU % Triple DES**
1	3.809	22.963
2	4.121	8.021
3	7.948	28.55
4	23.899	3.207
5	27.271	4.097
6	49.941	0.568
7	19.49817	18.924
AVERAGE	19.49817	12.33286

For clarification the results in the previous have been drawn figure 4.7. Figure 4.7 shows the CPU percentage for encryption algorithms DES and Triple DES with merge result hash and plaintext.

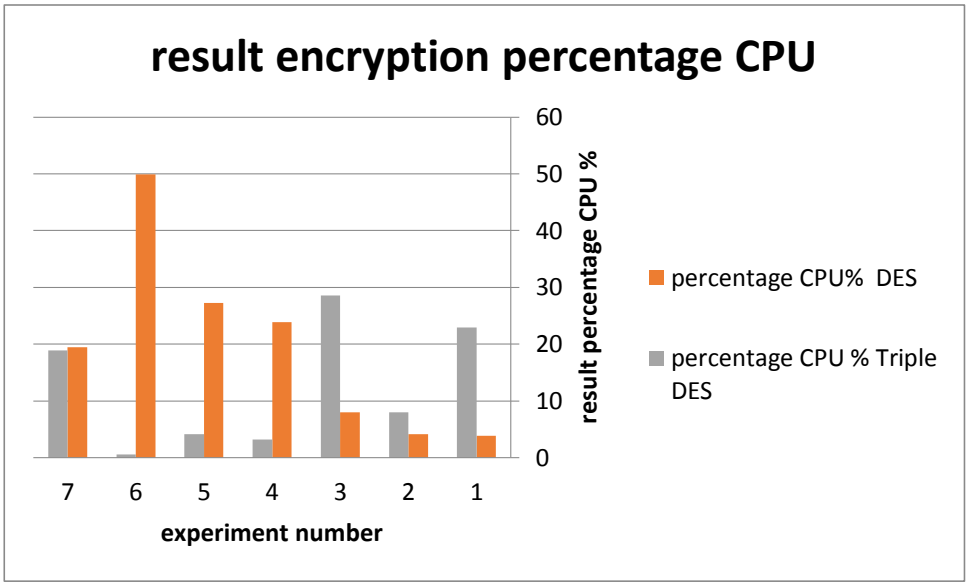


Figure4.7: encryption percentage CPU for merge

Figure4.7 shows that the experiments (1, 2 and 3) CPU percentage merge for DES is **less** than triple DES. Experiments (4, 5, 6 and 7) CPU percentage merge for Triple DES is less than DES. Figure 4.15 shows that the performance CPU for **Triple DES** is ***better*** than **DES**.

Figure 4.8, shows the average merge (MD5 plus plaintexts) percentage CPU for encryption.

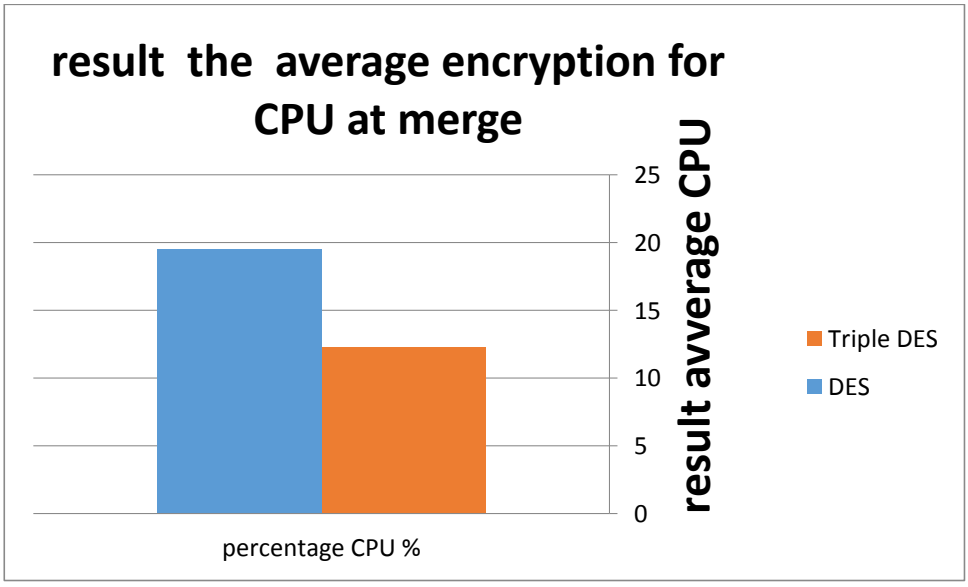


Figure 4.8: average the encryption percentage CPU for merge

From figure 4.8, it is clear average encryption execution time for DES and Triple DES. Figure 4.8shows that the performance average percentage CPU for **Triple DES is** better than **DES.** Since the lower CPU percentage is better than higher CPU percentage.

Table 4.10, result the merge MD5.

experiment number	**execution time (m.s) with DES**	**execution time (m.s) with Triple DES**
1	31	59
2	30	15
3	30	5
4	31	14
5	32	24
6	33	51
7	69	278
Average	36.6	63.71

For clarification the results in the previous table have been shown Figure 4.9. Figure 4.9, shows the encryption execution time for encryption algorithms DES and Triple DES with merge result hash and plaintext.

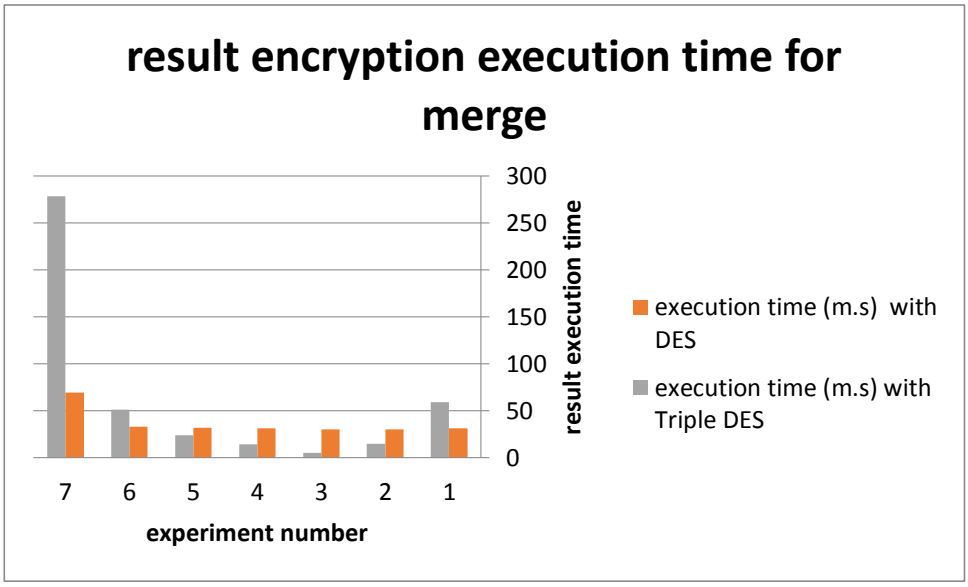


Figure4.9: Result sample merge MD5 for execution time.

Figure **4.9** that for the experiments (1, 6 and 7) execution time for **DES** is **less** than **Triple DES**.

Figure 4.10, shows the average encryption execution time for encryption algorithms DES and Triple DES with hash MD5.

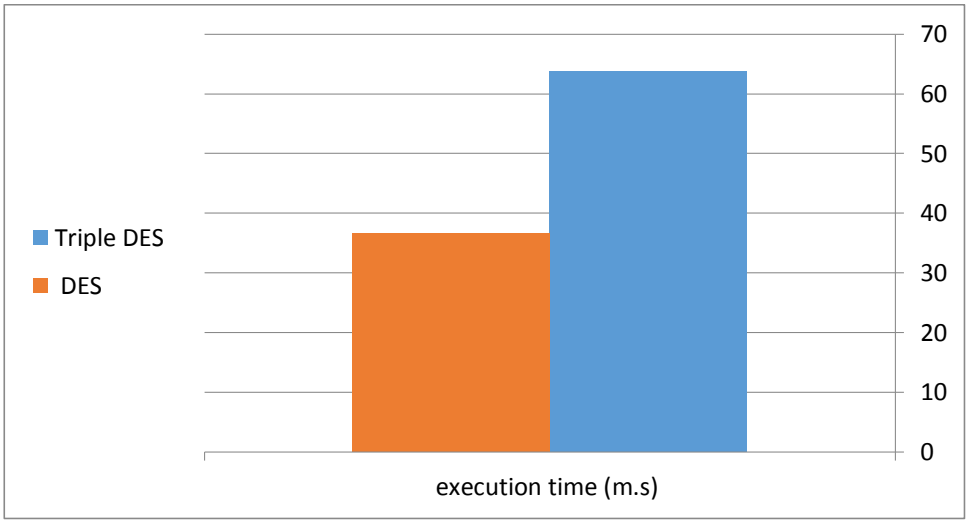


Figure4.10: Result sample merge MD5 for average execution time.

From figure 4.18, it is clear average encryption execution time for DES and Triple DES. Figure 4.10 shows that the performance average execution time for is **Triple DES <u>better</u>** than **DES.**

following the same way for calculating MD5, We continue further with the experimental results with the other two hashing namely, SHA-1, and SHA-256, to compare the result obtained scenario hashing over encrypted data in term of average duration time and CPU percentage(before and after) for SHA-1, and SHA256 algorithms and summarize their results in the following:

4.6SHA-1:

4.6.1 Summarize result hash (SHA-1) (encryption hashing):

Table4.23 shows summarize average result execution time and percentage CPU before encryption on scenario encryption hashing and scenario hashing over encrypted data.

Table 4.11summarize result before encryption

execution time before encryption scenario 1	CPU percentage before encryption scenario 1
456.6	31.72

Table 4.12: summarize result percentage CPU after encryption.

Summarize CPU percentage(after encryption hash)	**DES**	**Triple DES**
After encryption	72.72	72.72

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where DES equal Tripe DES.

Table 4.13: summarize result execution time after encryption with different keys.

Summarize **execution time (after encryption hash)**	**DES**	**Triple DES**
After encryption	90.73	90.73

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where DES with **ALL** better than Triple DES.

4.6.2 Summarize result merge hash (SHA-1) with plain text

Table 4.14: summarize result CPU percentage merge after encryption.

Summarize **CPU percentage (merge)**	**DES**	**Triple DES**
After encryption	82.26	82.39

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of

key size was very little and the accuracy of it was not good for that reasons, where DES **is better** than Triple DES.

Table 4.15: summarize result execution time merge after encryption with different keys.

Summarize execution time(m.s) (Merge).	**DES**	**Triple DES**
After encryption	72.72	1,284.47

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where DES is better than Triple DES.

4.7SHA-256:

4.7.1 Summarize result hash (SHA-256) for (encryption hashing):

Table 4.29shows summarize average result execution time and percentage CPU before encryption on scenario encryption hashing and scenario hashing over encrypted data.

Table 4.16summarize result before encryption

Execution time before encryption scenario 1	CPU percentage before encryption scenario 1
169.95	42.86

Table4.17: summarize result percentage CPU after encryption.

Summarize **CPU percentage**	**DES**	**Triple DES**
After encryption	**93.66**	85.93

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, Where Triple DES is better than and DES.

Table 4.18: summarize result execution time after encryption.

Summarize Execution time	**DES**	**Triple DES**
After encryption	3669.20	246.24

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where Triple DES for is better than DES.

4.7.2 Summarize result merge hash (SHA-256) with plain text:

Table 4.19: summarize result CPU percentage after encryption.

Summarize CPU percentage	DES	Triple DES
After encryption	87.50	89.01

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where DES **is** better than triple DES.

Table 4.20: summarize result execution time after encryption.

Summarize execution time.	DES	Triple DES
After encryption	1482.42	3874.07

The previous table, shows result hash MD5 after encryption with key size was one the parameter that we intended to investigate but we noted that some encryption techniques and hashing techniques did not allow us to choose the size of key. For that the effect of key size was very little and the accuracy of it was not good for that reasons, where Triple DES is **better** than DES.

We will compare the results obtained in MD5 used encryption algorithm with that obtained in SHA-1 and with SHA-256 in CPU percentage and execution time.

Table 4.21: compare between after encryption (percentage CPU).

Encryption Hashing	DES	Triple DES
MD5	9.37	10.79
SHA-1	72.72	72.72
SHA-256	**93.66**	85.93

Result in previous table which explain the result percentage CPU with hashing algorithms and encryption algorithms. Also these explained following:

- When the DES of SHA-256 increased than MD5 and SHA-1. We will Loss of performance DES = (93.66-9.37)/ 93.66= 89.995%
- When the Triple DES of SHA-256 increased than MD5 and SHA-1. We will Loss of performance Triple DES= (85.93-10.79)/ 85.93= 87.44%

Table 4.22 compare between after encryption (percentage CPU) merge with encryption for different merge hashing.

Encryption merge	DES	Triple DES
MD5	19.498	12.333
SHA-1	82.26	82.39
SHA-256	87.50	89.01

Result in previous table which explain the result percentage CPU with hashing algorithms and encryption algorithms. Also these explained following:

- When the DES of SHA-256 increased than MD5 and SHA-1. We will Lossof Loss performance DES = (87.50-19.498)/ 87.50= 77.72%.
- When the Triple DES of SHA-256 increased than MD5 and SHA-1. We will Loss performance Triple DES= (89.01-12.333)/89.01= 86.14%.

Table 4.23: compare between after encryption (Execution time) for different hashing.

Encryption hashing	DES	Triple DES
MD5	67.57	61.43
SHA-1	90.73	90.73
SHA-256	3669.20	246.24

Result in previous table which explain the result execution time with hashing algorithms and encryption algorithms. Also these explained following:

- When the DES of SHA-256 increased than MD5 and SHA-1. We will Loss performance DES = (3669.20-67.57)/ 3669.20= 98.19%.
- When the Triple DES of SHA-256 increased than MD5 and SHA-1. We will Loss performance Triple DES= (246.24-61.43)/246.24= 75.05%.

Table 4.24compare between after encryption (execution Time) for different merge hashing.

Encryption merge	DES	Triple DES
MD5	1,170.11	63.71
SHA-1	72.72	1,284.47
SHA-256	1482.42	3874.07

Result in previous table which explain the result **execution time** merge with hashing algorithms and encryption algorithms. Also these explained following:

- When the DES of SHA-256 increased than MD5 and SHA-1. We will Loss performance DES = (1482.42-72.72)/ 1482.42= 95.09%
- When the Triple DES of SHA-256 increased than MD5 and SHA-1. We will Loss performance Triple DES= (3874.07-63.71)/3874.07= 98.36%.

CHAPTER FIVE

Conclusion and Future Work

5.1 Introduction:

This section summarizes the conclusions of our work model and the proposed proposals for future work. In area 5.2 we present the principle conclusion from our model. While, section5.3. Is talking about some future headings and recommending acquiring changes the encryption techniques and hashing techniques to achieve security on the Cloud Computing.

5.2Conclusion:

This chapter concludes all about the performance for the two scenario, the performance in this research takes into consideration execution time and CPU percentage.

For the two approaches (before and after) the MD5 was the best. The following will conclude the most important results

1) ***Before encryptions:*** the average execution time and CPU percentage for all encryption are the same since all MD5 using the same key size.

2) After encryption: the following summarize the most important results.

- The Triple DES average execution time for all key size was the best.
- The DES average CPU percentage for all key size was the best.

Execution time all encryption is **higher** than before encryption.

5.3 Recommendations for Future Research

Through in this thesis; many ideas not achieved yet. We can suggest some ideas or future study:

- Using encryption algorithms other than the used in our related work in order to see different the execution in more optimal ways which addresses issues of time/ CPU and reduction cost.

- Using hashing algorithms other than the used in our related work in order to see different the execution in more optimal ways which addresses issues of time/ CPU and reduction cost.
- Study the performance (CPU & Time) when encryption runs on other application such as MATLAP.
- Execution the related work with applications which supports large key size instead of than keys use in this thesis.
- Finding the new performance parameter adding to CPU utilization and time execution such as RAM.

References

References:

- Al-Ahmad M. A. & I. F. Alshaikhli (2013). "Broad View of Cryptographic Hash Functions." *International Journal of Computer Science Issues journal (ISI Journal)*, 10(4).

- Al-Vahed, A. &Sahhavi, H. (2011), 'An overview of modern cryptography, World Applied Programming, 1(1), 55-61.

- Apostul, A., Pulcan, F., Ularu, G., Suciu, G. &Todoran G. (2013),'Study on advantages and disadvantages of Cloud'. *International journal of advanced research in computer science and software engineering*, 2(11), 200-205.

- Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, R., Lee, G., Patterson, D., Rabkin, A., Stoica, L. &Zaharia M. (2010),'A View of Cloud Computing',*Comm. ACM*, 53(4), 50–58.

- Ayushi. (2010), 'A Symmetric Key Cryptographic Algorithm', *International Journal of Computer Applications (0975 - 8887)*, 1(15), 1-4.

- Begum, R., Kumar, R. & Kishore, V. (2012), 'Data confidentiality scalability and Accountability (DCSA) in cloud computing'. *Recent Advances in Applied Computer Science and Digital Services*, 2(11), 118–123.

- Biham, E.,&Shamir A. (1991),'Differential cryptanalysis of DES-like cryptosystems', Journal of Cryptology, 4(1), 3-7.

- Bijwe, S.&Ramteke, P. (2015), Database in Cloud Computing - Database-as-a Service (DBaas) with its Challenges ', *International Journal of Computer Science and Mobile Computing*, 4(2), 73-79.
- Bisong A., &Rahman S. (2011), 'an overview of the security concerns in enterprise cloud computing, *international journal of network security &its applications(IJNSA)*, 3(1), 30-45.
- Bohn, R., Messina, J., Liu, F, Tong J. & Mao J. (2011),' NIST Cloud Computing Reference Architecture', 2011 IEEE World Congress on Services.
- Boldyreva, A., Chenette, N., Lee, Y. & O'Neill, A. (2009), 'Order preserving symmetric encryption', *In Proceedings of the 28th Annual International Conference on Advances in Cryptology*, EUROCRYPT '09, 224–241.
- Boldyreva, A ., Chenette, N., Lee, U.,& O'Neill, A. (2009), 'Order-Preserving Symmetric Encryption', *Georgia Institute of Technology, Atlanta, GA, USA*, 1-24.
- Bouganim, L., &Guo, Y., (2011), 'Database Encryption', *Encyclopedia of Cryptography and Security*'s, l (15), 1-9.
- Carroll, M. & Merwe, A. (2011),'Secure Cloud Computing Benefits, Risks and Controls', *Information Security South Africa (ISSA)*, 1-9.
- Chaeikar, S.& Jafari, M. (2012), 'Definitions and Criteria of CIA Security Triangle in Electronic Voting System',*International Journal of Advanced Computer Science and Information Technology (IJACSIT)*, 1(1), 14-24, October.

- Chou, T. (2013),' Security threats on cloud computing vulnerabilities', *International Journal of Computer Science & Information Technology (IJCSIT)*, 5(3), 79-88.

- Cioloca, C., &Georgesc M. (2011), 'Increasing Database Performance using Indexes', Database Systems Journal, II(2), 13-21.

- Cloud away. (2012), http://thecloudway.net/cloud-computing/deployment-models/.

- DataLossDB Open Security Foundation. http://datalossdb.org/statistics.

- Donkena, K. (2015), 'Performance Evaluation of Cloud Database and Traditional Database in terms of Duration time while Retrieving the Data', master thesis, Blekinge Institute of Technology, 371 79 Karlskrona.

- Durairaj, M.,&Kannan, P. (2014),' A Study On Virtualization Techniques And Challenges In Cloud Computing', *international journal of scientific & technology research*, 3(11), 147-151.

- Dwokin, M. (2001),'NIST Special Publication 800-38A Recommendations for Block Cipher Modes of Operation', *Methods and Techniques*, 1-59.

- Evdokimov, S.,&Gu¨nther, O.(2007), 'Encryption Techniques for Secure Database Outsourcing', *Biskup, J., Lopez, J. (Eds.)*, 4734, 1-23, Springer.

- Fanbao, L. and Feng, D., 2013. 'Fast Collision Attack on MD5', PP.5.

- Gawande, M., &Kapse, A. (2014),' Analysis of Data Confidentiality Techniques in Cloud Computing', *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 3(3), 169-175, ISSN: 2320-088X, Mar. 2014.

- Gentry, C. (2009), 'A fully homomorphic encryption scheme', *university in partial fulfillment of the requirement for the degree of doctor of philosophy*, 1-190, <https://crypto.stanford.edu/craig/craig-thesis.pdf>.

- Gonzalez, N., Miers, C., Red´ıgolo, F., s Simpl´ıcio, M., Carvalho, T., Naslund, M.,& Pourzandi, M. (2012), 'A quantitative analysis of current security concerns and solutions for cloud computing', *Gonzalez et al. Journal of Cloud Computing: Advances, Systems and Applications*, 1-18, 12 July, Springer.

- Gorelik, E (2013). 'Cloud Computing Models', Massachusetts Institute of Technology, 4-81, available on: <http://web.mit.edu/smadnick/www/wp/2013-01.pdf>.

- Gouda, k., Patro, A., Dwivedi, D., &Bhat, N. (2014), 'Virtualization Approaches in Cloud Computing**'**, International Journal of Computer Trends and Technology (IJCTT**)**, 12(4), 161-164.

- Gupta, P., &Kumar, P. (2014),'A comparative analysis of SHA and MD5 algorithm', *Piyush Gupta et al, / (IJCSIT) international journal of computer science and information technologies*, 5(3), 4492-4495.

- Hamlen, K., Kantarcioglum, M., Khan, L. &Thuraisingham, B. (2010),'Security issues for cloud computing', *international journal of information security and privacy*, 4(2), 39-51.

- Hashemi, S., & Hesarlo,P. (2014), 'Security, Privacy and Trust Challenges in Cloud Computing and Solutions', I.J. Computer Network and Information Security, 8, 34-40.

- Hashizume, K., Rosado, D., Medina, E., & Fernandez, E. (2013), 'An analysis of security issues for cloud computing', *Journal of Internet Services and Applications*, Springer, 1-13.

- Hodge, G. (2004), 'Computer-assisted database indexing: the state-of-the-art', The Indexer, 19(1), 23-27.

- NIST. (2011)., Available on: http://www.cloudcontrols.org/cloud-standard information/organizations/nist/sp500-292/.

- Karthik, S., &Muruganandam, A. (2014), ' data encryption and decryption by using triple DES and performance analysis of cryptosystem', international journal of scientific engineering and research (IJSER), issn (online): 2347-3878, 2(11), November 2014.

- Katz, J., &lindell, Y. (2007), 'Introduction to Modern Cryptography', Taylor and francis, Groub, an informa business, version date. 20110715, international standard number 978-1-58488-551-1, 1-529. *Master thesis*.

- Kaur, G., and Mahajan, M. (2013), 'Analyzing Data Security for Cloud Computing Using Cryptographic Algorithms', *Journal of Engineering Research and Applications ISSN : 2248-9622*, 3(5), Sep-Oct 2013, 782-786.

- Kaur, M., & Mahajan, M. (2013), 'Using encryption Algorithms to enhance the Data Security in Cloud Computing', *International Journal of Communication and Computer Technologies*, 01(12), 56-59.

- Koganti, K., Patnala, E., Narasingu, S.,&Chaitanya, J. (2013),'Virtualization Technology in Cloud Computing Environment', *International Journal of Emerging Technology and Advanced Engineering*, 3(3), 3-72.

- Kumar, M., Mishra, R., Pandey, R., & Singh, P. (2010), 'Comparing classical encryption with modern techniqes', *S-JPSET*, 1(1), 1-54.

- Makkar, L., and Rajput, G. (2013). 'Architecture and Security Functions of Cloud Computing', *International Journal of Computer Sciences and Management Researches*, 2(1), 1261-1264.

- Martinez, S., Miret, J., Tomas, R., and Valls, M. (2013), ' security analysis of order preserving symmetric cryptography', *applied mathematics& information an international journal*, 7(4), 11-14.

- Mell, P.,&Grance, T. (2011).'The NIST of Cloud Computing'. *NIST Special Publication 800-145*, National Institute of Standards and Technology NIST, Gaithersburg, MD, (September 2011).

- Nazir, M., Tiwari, P., Tiwari, S., &Mishra, R. (2015), 'Cloud Computing: An Overview', *Cloud Computing: Reviews, Surveys, Tools, Techniques and Applications – An Open-Access eBook by HCTL Open http://ebooks.hctl.orgISBN-13 (PDF): 978-1-62951-802-2*, 1-15.

- NIST, (2014), cryptography hash and SHA-3 standard development, (on-line), available: http://csrc.nist.gov/groups/ST/hash/index.html

- NIST. (2004), 'NIST Brief Comments on Recent Cryptanalytic Attacks on Secure Hashing Functions and Continued Security Provided by SHA-1', 25th August 2004, <http://csrc.nist.gov/groups/ST/toolkit/documents/shs/hash_standards_comments.pdf>.

- Paar, C., &Pelzl, J. (2010), 'understanding cryptography', *springer. Verlage berlin Heidelberg 2010*, ISBM: 978-3-642. 04100-6, e-ISBN: 978-3-642. 04101-3, Online: www.crypto-textbook.com.

- Pareek. (2012), ‘A Survey of Cryptographic based Security Algorithms for Cloud Computing have discussed the Fundamental Characteristics of Cloud Computing ', HCTL Open Int. J. of Technology Innovations and Research, 8, 1-17.

- Pareek. (2012), 'A design and analysis of a novel digital image encryption scheme', *International Journal of Network Security & Its Applications (IJNSA)*, 4(2), 95-108.

- Popa, R., Li H., &Zeldovich N. (2010), ' An ideal security protocol for ordering preserving encoding', *In Security and Privacy(SP)*,2013 IEEE Symposium.

- Prasanthi, O., & Reddy, M. S. (2012). 'RSA Algorithm Modular Multiplication'. *International Journal of Computer Applications in Engineering Sciences*, 2(2).

- Quebec, G. (2010),'Introduction to cloud computing', *Office of the privacy commissioner of Canada*, 1-6, <https://www.priv.gc.ca>.

- RajS.,Sharmila M., &BenetaP. (2013), Hybrid Cryptographic Processor for Secure Communication Using FPGA ' International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970), 3(4), 319-324.

- Rana, S., & Joshi, P. (2012),'Risk analysis in web application by using cloud computing', *international journal of multidisciplinary research*, 2(1), 386-394.

- Rivest, R. L., Shamir, A., &Adleman, L. (1977), 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems', *Communications of the ACM*, 21(2), 120-126, April.

- Roldan, F., Velasco, M.., &Llorente, G. (2012), 'Zero- padding or cyclic prefix for MDFT- based filter bank multicarrier communication', *Signal Processing An International Journal*, 106-117(12),1647-1657.

- Romine, C. (2012), 'Secure Hash Standard (SHS)', *federal information processing standards publications*, 3-31.

- Sangwan, N. (2012). 'Text Encryption with Huffman Compression', *International Journal of Computer Applications (0975 – 8887)*, 54(6), 29-32.

- Sarode, S., Giri, D. &Chopde, K. (2011), 'The Effective and Efficient Security Services for Cloud Computing', *International Journal of Computer Applications*, (0975 – 8887).

- Sehgal, B., &Narwal, J. (2015), 'An Analysis of Performance for Multi-Tenant Application through Cloud SIM', *International Journal of Emerging Research in Management &Technology*, 4(6), 179-138.

- Shankar, M., &akshaya, P. (2014), 'hybrid cryptographic technique using RSA algorithm and scheduling concepts ', *International Journal of Network Security & Its Applications (IJNSA)*, 6(6), 39-48.

- Shannon, C. (1984), 'A Mathematical Theory of Communication', *bell system technical journal*, 27, 5-83.

- Shi, Z., Ma, C., Cote, J., and Wang, B. (2012), 'Hardware Implementation of Hash Functions', viii(427), 27-50, Springer.

- Shmueli, E. Vaisenberg, R. Elovici, Y.& Glezer, C. (2009). 'Database Encryption – An Overview of Contemporary Challenges and Design Considerations', *Sig mod Record*, 38(3), 29-34.

- Singh, A., &Gilhotra, R. (2011). 'Iosr Data Security Using Private Key Encryption System Based on Arithmetic Coding', *International Journal of Network Security & Its Applications (IJNSA)*, 3(3), 58-67.

- Singh, S., Pandey, B., Srivastava, R., Rawat, N., Rawat, P., &Awantika. (2014), 'Cloud Computing Attacks: A Discussion with solutions', *open journal of mobile computing and cloud computing*, 1(1), 1-8.

- Singla, S.,& Singh, J. (2013),'Survey on Enhancing Cloud Data Security using EAP with Rijndael Encryption Algorithm', *Global Journal of Computer Science and Technology Software & Data Engineering*, 13(5), 11-14.

- Singth, S., Maakar, S., &Kumar, S. (2013), 'a performance analysis of DES and RSA cryptography', *international journal of emerging trends& technology in computer science (IJETTCS)*, 2(3), 418-423.

- Sobti R., &Geetha G. (2012), 'Cryptographic Hash Functions: A Review', *IJCSI International Journal of Computer Science Issues*, 9(2), 461-479.

- Soofi, A., Khan, M.,Talib, R. & Sarwar, U. march 2014. Security Issues in Saas Delivery Model of Cloud Computing, *International Journal of computer science and mobile computing (IJCSMC)*, 3(3), 15-21.

- Staff, R. (2014), 'Securing the cloud with homomorphic encryption', *The Next Wave*, 20(3), 1-4.

- Stehlé, D., &Steinfeld, R. (2010). 'Faster fully homomorphic encryption'. *InAdvances in Cryptology-ASIACRYPT*.

- Stevens, S. (2007), 'On Collisions for MD5', *Eindhoven University of Technology Department of and Computing Science*, Springer Berlin Heidelberg.

- Sunitha, K., &Prashanth, S. (2013), 'A Study of Encryption Algorithms AES, DES and RSA for Security', *Global Journal of Computer Science and Technology*, XIII (XV), version.1.0, 15-22, Online ISSN: 0975-4172 & Print ISSN: 0975-4350.

- Teebaa,M.,&Hajii, S. (2013). 'Secure cloud computing through homomorphic encryption', *international journal of advancement in computing technology (IJACT)*, 5(16), 29-38.

- Tiwari, H., &Asawa, K. (2010), 'A Secure Hash Function MD-192 with modified message expansion', *(IJCSIS) International Journal of Computer Science and Information Security*,VII (II), 108-111.

- Vijayaraj, A., M Ram, S. (November 2011),'Analysis of the characteristics and trusted security of cloud computing', *International Journal on Cloud Computing: Services and Architecture*, 1(3), 61-69.

- WaterOx. (2013). 'Indexing encrypted data', *indexing encrypted data-sql server-Toad world*, 1-10.

- Woodbury,C. (2007), 'Your Guide to a Successful Encryption Project', *Security - IBM i (OS/400, i5/OS)*, available on: http://www.mcpressonline.com/security/ibm-i-os400-i5os/your-guide-to-a-successful-encryption-project.html.

- Wu, S., Jiang, G., Ooi, B .,& Wu, K. (2010), 'Efficient B-tree Based Indexing for Cloud Data Processing', *Proceedings of the VLDB Endowment*, 3(1),1207-1218.

- Zunnurhain, K., &Vrbsky, S. (2010), 'Security Attacks and Solutions in Clouds', *Department of Computer Science the university of Alabama Tuscaloosa, al 35487-0290*, pp. 1-4,Available on:

  http://salsahpc.indiana.edu/CloudCom2010/Poster/cloudcom2010_submission_98.pdf.

- Al Shehri, W. (2013), 'Cloud Database AS Services' Data base as service', International Journal of Database Management Systems (IJDMS), 5(2), 1-12.

Appendix I:

1- Steps before Encryption hashing

Table 4.1 MD5 experiment results for before encryption DES, 3DES and RSA.

Exp. No	**FileSize**	**Hash name**	**Size hash**	**CPU%**	**Time (m.s)**	**Name Enc**
1	10 K byte	MD5	32	1.190	49	DES
1	10 K byte	MD5	32	1.190	49	3DES
1	10 K byte	MD5	32	1.190	49	DES
1	10 K byte	MD5	32	1.190	49	3DES
1	10 K byte	MD5	32	1.190	49	DES
1	10 K byte	MD5	32	1.190	49	3DES
2	10K byte	MD5	32	2.623	11	DES
2	100 K byte	MD5	32	2.623	11	3DES
2	100 K byte	MD5	32	2.623	11	DES
2	100 K byte	MD5	32	2.623	11	3DES
2	100 K byte	MD5	32	2.623	11	DES
2	100 K byte	MD5	32	2.623	11	3DES
3	500 K byte	MD5	32	1.318	3	DES
3	500 K byte	MD5	32	1.318	3	3DES
3	500 K byte	MD5	32	1.318	3	DES
3	500 K byte	MD5	32	1.318	3	3DES
3	500 K byte	MD5	32	1.318	3	DES
3	500 K byte	MD5	32	1.318	3	3DES
4	1500 500 K byte	MD5	32	1.727364	10	DES
4	1500 500 K byte	MD5	32	1.727364	10	3DES
4	1500 500 K byte	MD5	32	1.727364	10	DES
4	1500 500 K byte	MD5	32	1.727364	10	3DES
4	1500 500 K byte	MD5	32	1.727364	10	DES
4	1500 500 K byte	MD5	32	1.727364	10	3DES
5	1500 500 K byte	MD5	32	0.4100	20	DES
5	1500 500 K byte	MD5	32	0.4100	20	3DES

5	1500 500 K byte	MD5	32	0.4100	20	DES
5	1500 500 K byte	MD5	32	0.4100	20	3DES
5	1500 500 K byte	MD5	32	0.4100	20	DES
5	1500 500 K byte	MD5	32	0.4100	20	3DES
6	10 M byte	MD5	32	1.482	47	DES
6	10 M byte	MD5	32	1.482	47	3DES
6	10 M byte	MD5	32	1.482	47	DES
6	10 M byte	MD5	32	1.482	47	3DES
6	10 M byte	MD5	32	1.482	47	DES
6	10 M byte	MD5	32	1.482	47	3DES
7	100 M byte	MD5	32	21.519	274	DES
7	100 M byte	MD5	32	21.519	274	3DES
7	100 M byte	MD5	32	21.519	274	DES
7	100 M byte	MD5	32	21.519	274	3DES
7	100 M byte	MD5	32	21.519	274	DES
7	100 M byte	MD5	32	21.519	274	3DES

Where:

- Exp No: Experiment Number.
- File Size: The size of the file.
- Hash name: The name of hash
- Time: execution of result hash before encryption task, measured by minutes and seconds.
- CPU: execution of result hash before encryption task, measured percentage.

2- Steps before Encryption hashing (SHA-1)

Exp. No	e FileSize	Hash name	Size hash	CPU%	Time (m.s)	Name Enc
1	10 K byte	SHA-1	40	16.96	4	DES
1	10 K byte	SHA-1	40	16.96021	4	3DES
1	10 K byte	SHA-1	40	16.96	4	DES
1	10 K byte	SHA-1	40	16.96021	4	3DES
1	10 K byte	SHA-1	40	16.96021	4	DES
1	10 K byte	SHA-1	40	16.96021	4	3DES
2	10K byte	SHA-1	40	10.71005	1	DES
2	100 K byte	SHA-1	40	10.71005	1	3DES
2	100 K byte	SHA-1	40	10.71	1	DES
2	100 K byte	SHA-1	40	10.71005	1	3DES
2	100 K byte	SHA-1	40	10.71005	1	DES
2	100 K byte	SHA-1	40	10.71005	1	3DES
3	500 K byte	SHA-1	40	13.71437	6	DES
3	500 K byte	SHA-1	40	13.71437	6	3DES
3	500 K byte	SHA-1	40	13.71	6	DES
3	500 K byte	SHA-1	40	13.71437	6	3DES
3	500 K byte	SHA-1	40	13.71437	6	DES
3	500 K byte	SHA-1	40	13.71437	6	3DES
4	1500 500 K byte	SHA-1	40	30.75195	20	DES
4	1500 500 K byte	SHA-1	40	30.75195	20	3DES
4	1500 500 K byte	SHA-1	40	30.75	20	DES
4	1500 500 K byte	SHA-1	40	30.75195	20	3DES
4	1500 500 K byte	SHA-1	40	30.75195	20	DES
4	1500 500 K byte	SHA-1	40	30.75195	20	3DES
5	1500 500 K byte	SHA-1	40	51.32312	58	DES
5	1500 500 K byte	SHA-1	40	51.32312	58	3DES
5	1500 500 K byte	SHA-1	40	51.32	58	DES
5	1500 500 K byte	SHA-1	40	51.32312	58	3DES
5	1500 500 K byte	SHA-1	40	51.32312	58	DES

5	1500 500 K byte	SHA-1	40	51.32312	58	3DES
6	10 M byte	SHA-1	40	63.54527	128	DES
6	10 M byte	SHA-1	40	63.54527	128	3DES
6	10 M byte	SHA-1	40	63.55	128	DES
6	10 M byte	SHA-1	40	63.54527	128	3DES
6	10 M byte	SHA-1	40	63.54527	128	DES
6	10 M byte	SHA-1	40	63.54527	128	3DES
7	100 M byte	SHA-1	40	61.33653	1,160.00	DES
7	100 M byte	SHA-1	40	61.33653	1,160.00	3DES
7	100 M byte	SHA-1	40	61.34	1,160.00	DES
7	100 M byte	SHA-1	40	61.33653	1,160.00	3DES
7	100 M byte	SHA-1	40	61.33653	1,160.00	DES
7	100 M byte	SHA-1	40	61.33653	1,160.00	3DES

3- Steps before Encryption hashing (SHA-256)

Exp. No	e FileSize	Key Size (bit)	Hash name	Size hash	CPU%	Time (m.s)	Name Enc
1	10 K byte	10	SHA-256	40	45.88	87	DES
1	10 K byte	10	SHA-256	40	45.88	87	3DES
1	10 K byte	100	SHA-256	40	45.88018	87	DES
1	10 K byte	100	SHA-256	40	45.88	87	3DES
1	10 K byte	200	SHA-256	40	45.88	87	DES
1	10 K byte	200	SHA-256	40	45.88018	87	3DES
2	10K byte	10	SHA-256	40	36.75137	4	DES
2	100 K byte	10	SHA-256	40	36.75	4	3DES
2	100 K byte	100	SHA-256	40	36.75	4	DES
2	100 K byte	100	SHA-256	40	36.75137	4	3DES
2	100 K byte	200	SHA-256	40	36.75	4	DES
2	100 K byte	200	SHA-256	40	36.75	4	3DES
3	500 K byte	10	SHA-256	40	40.41634	20	DES
3	500 K byte	10	SHA-256	40	40.42	20	3DES
3	500 K byte	100	SHA-256	40	40.42	20	DES
3	500 K byte	100	SHA-256	40	40.41634	20	3DES
3	500 K byte	200	SHA-256	40	40.42	20	DES
3	500 K byte	200	SHA-256	40	40.42	20	3DES
4	1500 500 K byte	10	SHA-256	40	39.22281	53	DES
4	1500 500 K byte	10	SHA-256	40	39.22	53	3DES
4	1500 500 K byte	100	SHA-256	40	39.22	53	DES
4	1500 500 K byte	100	SHA-256	40	39.22281	53	3DES

4	1500 500 K byte	200	SHA-256	40	39.22	53	DES
4	1500 500 K byte	200	SHA-256	40	39.22	53	3DES
5	1500 500 K byte	10	SHA-256	40	54.59304	103	DES
5	1500 500 K byte	10	SHA-256	40	54.59	103	3DES
5	1500 500 K byte	100	SHA-256	40	54.59	103	DES
5	1500 500 K byte	100	SHA-256	40	54.59304	103	3DES
5	1500 500 K byte	200	SHA-256	40	54.59	103	DES
5	1500 500 K byte	200	SHA-256	40	54.59	103	3DES
6	10 M byte	10	SHA-256	40	41.03634	347	DES
6	10 M byte	10	SHA-256	40	41.04	347	3DES
6	10 M byte	100	SHA-256	40	41.04	347	DES
6	10 M byte	100	SHA-256	40	41.03634	347	3DES
6	10 M byte	200	SHA-256	40	41.04	347	DES
6	10 M byte	200	SHA-256	40	41.04	347	3DES
7	100 M byte	10	SHA-256	40	35.94444	3,821.00	DES
7	100 M byte	10	SHA-256	40	35.94	3,821.00	3DES
7	100 M byte	100	SHA-256	40	35.94	3,821.00	DES
7	100 M byte	100	SHA-256	40	35.94444	3,821.00	3DES
7	100 M byte	200	SHA-256	40	35.94	3,821.00	DES
7	100 M byte	200	SHA-256	40	35.94	3,821.00	3DES

4- After Encryption hashing (md5)

Name Enc	Encryption hashing size	Enc TimeHashing (m.s)	CPU encryption hashing Enc
DES	48	96	2.00
3DES	32	54	23.33
DES	48	82	1.23
3DES	32	51	6.45
DES	48	87	1.499
3DES	32	51	1.606
DES	40	13	3.56
3DES	32	13	9.16
DES	40	12	2.76
3DES	32	12	6.45
DES	40	12	2.678
3DES	32	14	3.167
DES	40	5	2.15
3DES	32	4	10.44
DES	40	4	1.87
3DES	32	5	2.24
DES	40	4	1.739
3DES	32	5	2.382
DES	40	12	2.093
3DES	32	12	3.08
DES	40	11	2.03
3DES	32	12	1.59
DES	40	11	2.444
3DES	32	12	1.891
DES	40	24	3.064
3DES	32	22	1.895
DES	40	21	2.04
3DES	32	22	1.59
DES	40	21	1.485
3DES	32	22	1.071
DES	40	49	2.79
3DES	32	49	1.91
DES	40	48	1.78
3DES	32	49	2.796
DES	40	48	2.718
3DES	32	48	8.016

DES	40	274	49.945
3DES	32	276	25.699
DES	40	274	22.22
3DES	32	276	27.695
DES	40	275	23.114
3DES	32	275	26.61

5- After Encryption hashing (SHA-1)

Name Enc	Encryption hashing size	Enc TimeHashing (m.s)	CPU encryption hashing Enc
DES	48	38	75.313
3DES	40	6	59.753
DES	48	39	31.588
3DES	40	243	45.33
DES	48	37	77.928
3DES	40	6	70.499
DES	48	3	75.44
3DES	40	3	75.088
DES	48	1	22.333
3DES	40	11	27.603
DES	48	3	70.497
3DES	40	4	74.13
DES	48	9	75.33
3DES	40	8	70.779
DES	48	7	26.106
3DES	40	58	33.006
DES	48	7	77.27
3DES	40	8	75.706
DES	48	22	99.39
3DES	40	22	57.525
DES	48	21	43.579
3DES	40	27	79.76
DES	48	21	100.435
3DES	40	22	38.581
DES	48	59	70.74
3DES	40	105	62.445
DES	48	59	116.458
3DES	40	60	100.225
DES	48	59	57.882
3DES	40	112	75.755
DES	48	130	84.04
3DES	40	130	125.955
DES	48	164	106.51

3DES	40	168	74.753
DES	48	129	124.884
3DES	40	130	100.535
DES	48	1,162.00	81.06
3DES	40	1,162.00	117.602
DES	48	1,185.00	104.468
3DES	40	1,220.00	125.538
DES	48	1,161.00	77.247
3DES	40	1,162.00	119.299

6- After Encryption hashing (SHA-256)

Name Enc	Encryption hashing size	Enc TimeHashing (m.s)	CPU encryption hashing Enc
DES	72	342	93.95
3DES	64	88	98.76
DES	72	101	98.632
3DES	64	89	86.272
DES	72	114	97.079
3DES	64	89	90.429
DES	72	18	81.95
3DES	64	6	71.94
DES	72	5	74.027
3DES	64	5	87.688
DES	72	6	82.344
3DES	64	6	71.792
DES	72	20	80.94
3DES	64	22	83.27
DES	72	21	79.382
3DES	64	43	78.846
DES	72	21	76.508
3DES	64	22	77.294
DES	72	55	76.23
3DES	64	72	74.71
DES	72	54	77.776
3DES	64	55	91.384
DES	72	54	75.054
3DES	64	106	83.81
DES	72	104	99.46
3DES	64	106	98.24
DES	72	104	108.758
3DES	64	105	88.738
DES	72	104	91.218
3DES	64	105	94.209
DES	72	347	81.81
3DES	64	348	98.97
DES	72	348	77.784

3DES	64	349	78.696
DES	72	348	80.383
3DES	64	348	89.486
DES	72	3,822	78.94
3DES	64	3,887	85.87
DES	72	3,822	76.6
3DES	64	4,351	77.125
DES	72	3,822	69.963
3DES	64	3,822	80.371

7- After hashing over encrypted data (MD5)

Encryption name	**Encryption size merge**	**Time**	**CPU**
DES	88	**3,138**	3.809
3DES	80	59	22.963
DES	88	3,086	0.979
3DES	80	53	0.831
DES	88	3,091.00	1.25
3DES	80	53	22.486
DES	88	**3,023**	4.121
3DES	80	15	8.021
DES	88	3,022	0.543
3DES	80	13	23.38
DES	88	3,022	1.107
3DES	80	17	1.096
DES	88	**3,045**	7.948
3DES	80	5	28.55
DES	88	3,037	3.453
3DES	80	7	1.738
DES	88	3,036.00	3.049
3DES	80	7	2.138
DES	88	**3,067**	23.899
3DES	80	14	3.207
DES	88	3,071	15.605
3DES	80	14	1.971
DES	88	3,136.00	9.211
3DES	80	24	0.451
DES	88	**3,134**	27.271
3DES	80	24	4.097
DES	88	3,119	25.968
3DES	80	24	3.949
DES	88	3,136.00	16.34
3DES	80	24	0.706
DES	88	**3,360.00**	49.941
3DES	80	51	0.568
DES	88		25.968

		3,354	
3DES	80	51	3.949
DES	88	3,372.00	26.668
3DES	80	51	28.949
DES	88	**6,932.00**	19.49817
3DES	80	278	18.924
DES	88	6,406	22.062
3DES	80	278	9.06
DES	88	6,353.00	27.69
3DES	80	276	26.754

8- After hashing over encrypted data (SHA-1)

Encryption name	**Encryption size merge**	**Time**	**CPU**
DES	88	3,042.00	124.9
3DES	80	8	99.527
DES	88	3,043.00	13.966
3DES	80	482	63.227
DES	88	3,041.00	122.039
3DES	80	8	85.102
DES	88	3,014.00	129.33
3DES	80	5	109.592
DES	88	3,011.00	26.242
3DES	80	21	67.381
DES	88	3,013.00	123.985
3DES	80	7	82.472
DES	88	3,049.00	130.63
3DES	80	10	83.78
DES	88	3,047.00	28.937
3DES	80	110	73.494
DES	88	3,046.00	130.754
3DES	80	10	83.91
DES	88	3,111.00	135.83
3DES	80	24	81.92
DES	88	3,099.00	51.493
3DES	80	34	71.227
DES	88	4,036.00	97.329
3DES	80	24	72.294
DES	88	3,189.00	40.69
3DES	80	152	64.476
DES	88	3,213.00	139.103
3DES	80	62	82.392
DES	88	3,201.00	30.455
3DES	80	166	62.095
DES	88	3,541.00	91.99
3DES	80	132	71.424
DES	88	4,100.00	71.508

3DES	80	208	70.076
DES	88	3,583.00	143.506
3DES	80	132	88.835
DES	88	9,671.00	63.49
3DES	80	1,164.00	74.832
DES	88	7,803.00	113.066
3DES	80	1,280.00	89.501
DES	88	7,203.00	86.106
3DES	80	1,164.00	85.847

9- After hashing over encrypted data (SHA-256)

Encryption name	**Encryption size merge**	**Time**	**CPU**
DES	136	3,377.00	100.05
3DES	128	89	107.333
DES	136	3,106.00	93.67
3DES	128	91	75.05
DES	136	3,116.00	90.94
3DES	128	91	84.94
DES	136	3,022.00	83.9
3DES	128	8	74.664
DES	136	3,021.00	77.39
3DES	128	6	89.63
DES	136	3,011.00	83.51
3DES	128	8	75.95
DES	128	24	99.64
3DES	128	58	89.79
DES	128	66	72.67
3DES	128	80	82.27
DES	128	24	84.82
3DES	128	24	85.08
DES	136	3,228.00	81.19
3DES	128	91	77.616
DES	136	3,126.00	96.89
3DES	128	57	91.43
DES	136	3,144.00	82.83
3DES	128	159	88.86
DES	136	3,501.00	97.11
3DES	128	109	80.392
DES	136	3,502.00	108.22
3DES	128	107	74.94
DES	136	3,196.00	85.86
3DES	128	107	72.45
DES	136	3,672.00	100.68
3DES	128	349	94.738
DES	136	7,946.00	90.39

3DES	128	351	81.03
DES	136	3,675.00	95.23
3DES	128	349	83.54
DES	136	14,391.00	95.49
3DES	128	3,953.00	114.631
DES	136	11,830.00	94.19
3DES	128	4,881.00	80.27
DES	136	11,882.00	98.64
3DES	128	3,823.00	99.73

10- After hashing over encrypted data (SHA-256)

Encryption name	**Encryption size merge**	**Time**	**CPU**
DES	136	3,377.00	100.05
3DES	128	89	107.333
DES	136	3,106.00	93.67
3DES	128	91	75.05
DES	136	3,116.00	90.94
3DES	128	91	84.94
DES	136	3,022.00	83.9
3DES	128	8	74.664
DES	136	3,021.00	77.39
3DES	128	6	89.63
DES	136	3,011.00	83.51
3DES	128	8	75.95
DES	128	24	99.64
3DES	128	58	89.79
DES	128	66	72.67
3DES	128	80	82.27
DES	128	24	84.82
3DES	128	24	85.08
DES	136	3,228.00	81.19
3DES	128	91	77.616
DES	136	3,126.00	96.89
3DES	128	57	91.43
DES	136	3,144.00	82.83
3DES	128	159	88.86
DES	136	3,501.00	97.11
3DES	128	109	80.392
DES	136	3,502.00	108.22
3DES	128	107	74.94
DES	136	3,196.00	85.86
3DES	128	107	72.45
DES	136	3,672.00	100.68
3DES	128	349	94.738
DES	136	7,946.00	90.39

3DES	128	351	81.03
DES	136	3,675.00	95.23
3DES	128	349	83.54
DES	136	14,391.00	95.49
3DES	128	3,953.00	114.631
DES	136	11,830.00	94.19
3DES	128	4,881.00	80.27
DES	136	11,882.00	98.64
3DES	128	3,823.00	99.73