An Enhanced Technique for Segmentation and Encrypt Data to Storing in Cloud Computing

تقنية محسنة لتجزئة وتشفير البيانات لتخزينها في الحوسبة السحابية

By

Talal Abdulkarem Al-dhaheri

Supervised By

Dr. Hazim A. Farhan

Master Thesis

Submitted in Partial Fulfillment of the Requirements for the Master Degree In Computer Information System

Department of Computer Information System

Faculty of Information Technology

Middle East University

Amman – Jordan

July, 2013

Middle East University

Authorization Statement

I, Talal Abdulkarem Al-dhaheri, 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 regulations.

Name: Talal Abdulkarem Al-dhaheri

Date: 28/07/2013

Signature 

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "**An Enhanced Technique for Segmentation and Encrypt Data to Storing in Cloud Computing** " was successfully defend and approved on July 28 2013.

Examination Committee Member		Signature
1- Dr. Ahmad A. Kayed Associate Professor Chair of Department of Computer Science Faculty of Information Technology Middle East University	Chairman	.....
2- Prof. Nael B. Hirzallah Professor Dean of Faculty of Information Technology Applied Science Private University	Member	.....
3- Dr. Hazim A. Farhan Associate Professor Department of Computer Science Faculty of Information Technology Alzaytoonah University of Jordan	Supervisor and Member	.....

Middle East University

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 the thank and love to my parents for their encouragement, their moral support me in my study.

I would like also to express my deepest gratitude and appreciation to my thesis supervisor Dr. Hazim A. Farhan for his effort, helpful instructions, continues guidance, support, enthusiasm and inspiration during the work on 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. And last but certainly not least, I would like to express my deepest love and gratitude to my wife, my brothers and sisters for their unlimited encouragement, valuable support and patience throughout my study.

Middle East University

Dedication

I dedicate this work to my father, my mother, my partner (my wife), my brothers and sisters. For their patience, understanding, support, and most of all love, the completion of this work would not have been possible. I also dedicate to whoever taught me a letter on my way of knowledge.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

List of Abbreviations

Abbreviation	Description
SAAS	Software As A Service
PAAS	Platform As A Service
IAAS	Infrastructure As A Service
VM	Virtual Machine
EC2	Elastic Compute Cloud
AES-256	Advanced Encryption Standard
S3	Simple Storage Service
API	Application Programming Interface
DOS	Denial of Service
SSL	Secure Socket Layer
RSA	Rivest-Shamir-Adleman
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
Chapter3	
DSA	Data-Segmentation Algorithm
HSA	Hybrid Segmentation Algorithm
SR_i	Segment number
L_j	Level number
G_n	Group number
DSA	Database Segmentation Algorithm
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
MIPS-years	millions of instructions per second times the required number of years

Abstract

In the recent few years Cloud Computing has grown significantly, it became an important area of research and one of the vitality areas in the redevelopment of the infrastructure of information technology**.** Cloud Computing has been classified into several models based on the type of service provided to customers**.** Cloud Computing has many benefits, as well as, it has many challenges and concerns such as: security, privacy, data integrity, protection of property rights and other problems that make users and organizations fear to dealing with it**.** The main problem in Cloud Computing is the security of data**.** Therefore, this thesis came to present a new technique to encrypt and store the data in the Cloud Computing. The new technique divides the data into three levels according to their importance depending on the point of view of the data owner. The data in each level can be encrypted by using different algorithms and keys before store them in the Cloud. Our new technique aim to store data in a secure and safe way in order to avoid intrusions and attacks. Also, it will reduce the cost and time to store the encrypted data in the Cloud Computing. We are conducting a performance analysis by implementing the Elliptic Curve Cryptography (ECC) in all levels in order to check the performance of our model. Finally, the results of performance evaluation and comparing with other used techniques, we noted that our new technique enhance the trust in the Cloud Computing.

المـلخـص

شهدت السنوات الأخيرة تزايدا" في أنتشار الحوسبة السحابية ونموها بخطوات متسارعة حتى أصبحت مجالا هاما للبحث و واحدة من أكثر المناطق حيوية في إعادة تطوير البنية التحتية لتكنولوجيا المعلومات. تم تصنيفها إلى عدة نماذج أعتمادا" على نوع الخدمة المقدمة للعملاء. الحوسبة السحابية لها فوائد عديدة، وكذلك لديها بعض من التحديات والمخاوف مثل: الأمن والخصوصية وسلامة البيانات، وحماية حقوق الملكية وغيرها من التحديات التي تجعل المستخدمين والمنظمات يتخوفوا من التعامل معها. المشكلة الرئيسية في شبكات الحوسبة السحابية هي أمن البيانات. لذلك، جاءت هذه الأطروحة لتقديم تقنية جديدة للتشفير وتخزين البيانات في الحوسبة السحابية. تحتوي هذه التقنية على نظامين فرعيين هما نظام يقسم البيانات إلى ثلاثة مستويات وفقا لأهميتها من وجهة نظر مالك البيانات. أما النظام الفرعي الثاني فهو نظام تشفير البيانات في كل مستوى باستخدام خوارزميات ومفاتيح تشفير مختلفة قبل تخزينها في السحابة. تهدف هذة الرسالة الى تخزين البيانات بطريقة آمنة ومحمية من أجل تجنب الاختراقات والهجمات. أيضا، لتقلل التكلفة والوقت الألزم لتخزين البيانات المشفرة في الحوسبة السحابية. عملنا إجراء تحليل الأداء من خلال تنفيذ تشفير المنحنى الاهليلجيه في عدة مستويات. وأخيرا، أجرينا دراسة مقارنة لتقييم نتائج نموذجنا مع نتائج التقنيات المستخدمة الأخرى، لاحظنا أن أسلوبنا الجديد من شأنه أن يعزز الثقة في الحوسبة السحابي .

Chapter One

Introduction

Chapter One

Introduction and Hypothesis of the Study

1.1 Introduction

The development and growth of information and communication technology in this accelerating form make it the major target and the most sensitive dimension for many researchers. Studies in this field show an increasing in the rate of attacks and attempted attack either to obtain information or destroyed it. Fully secure and protected information exchange environment is still the goal which has not been achieved, and it needs a many of research and studies.

In the recent years, Cloud Computing has grown significantly to become one of the important areas of research (Table 1.1 presents the growth of Cloud applications during the last years). Many organizations in all fields are using it in several ways, and researchers are continuously work to develop and improve Cloud Computing environments, that shown in figure 1. Although Cloud Computing provides the best way to improve information technology resources, reduce costs and increase flexibility and efficiency, but the term Cloud is a concept that is still the process of research and is not clearly defined. There are many definitions, such as, “Cloud Computing is a model of delivering Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) in which the customer pays to use, rather than own, computational resources. It is particular suited to shared service delivery” (Plummer et al., 2008). Another definition of Cloud Computing is introduced by (NIST, 2011) as “Cloud describes the use of a collection of services, applications, information, and infrastructure comprised of pools of computer, network, information and storage resources. These components can be rapidly orchestrated, provisioned, implemented and

decommissioned, and scaled up or down providing for an on-demand utility-like model of allocations and consumption".

Table 1.1: Proportion average IT services and applications in the Cloud in 2011, 2012 and expectations for 2014(Cisco,2012)

sectors	2011	2012	Ideal in 2 Years
Retail	7%	36%	58%
Service Providers	13%	34%	53%
Finance	8%	31%	54%
Government	5%	28%	52%
Healthcare	3%	25%	44%

In recent years, the Cloud has been classified into several models based on the type of service provided to its customers as shown in Figure 1.1 and Figure 1.2 .Cloud services can be used in a private, public, and community/managed or hybrid setting. Although the Cloud Computing has many benefits, but it has some challenges and concerns that want many of the aspects trust by customers whose using it.

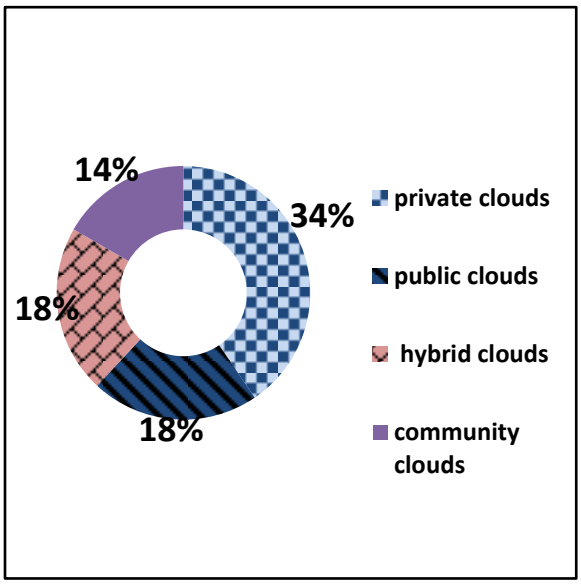

Figure 1.1:The Cloud Computing Models in 2011 (Cisco,2012)

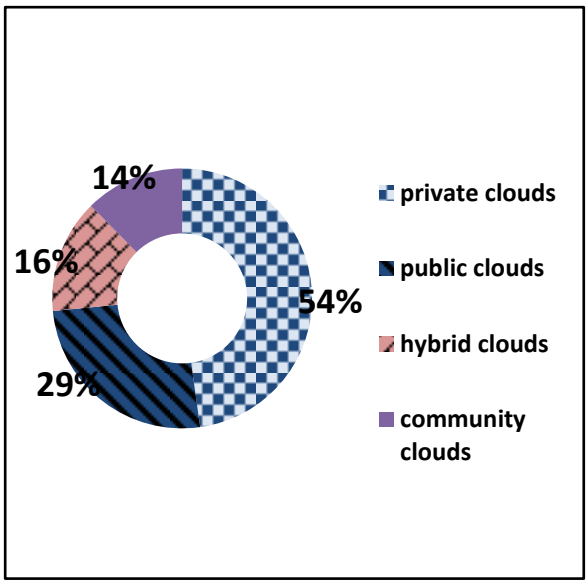

Figure 1.2:The Cloud Computing Models in 2012 (Cisco,2012)

Cloud Computing offers concerns and untrusted such as security, privacy, data integrity, protection of property rights and other problems that make users and organizations fear to dealing with Cloud Computing as shown in Figure 1.3. "attackers focus on how to impact the operations of other Cloud customers, and how to gain unauthorized access to data" (Cloud Security Alliance 2010: 11). Several studies had been proposed and developed by organizations and researchers to solve these challenges and concerns to gain user confidence in dealing with Cloud Computing. One of solutions within the Cloud work Dropbox specific to each user, Dropbox is one of the most popular online file saving services available (Wang, et al., 2012). Users can store data in an online system and then access that data from any location with internet access, also there is another solution to meet the challenges of Cloud Computing such as a Virtual Machine (VM) is "the software implementation of a computer that runs its own operating system and application as if it was a physical machine) "VMWare 2009). Another solution to enhance the security and privacy within the Cloud Computing is the use of encryption, where the owner of the data performs the encryption of the data and then sent to be stored in the Cloud Computing.

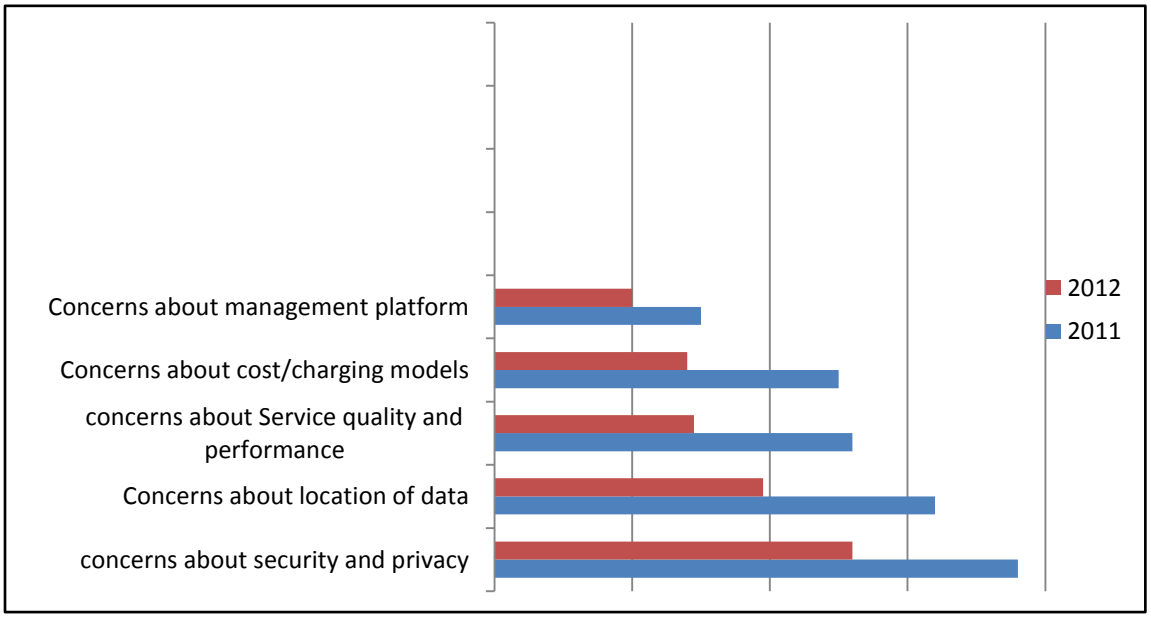

Figure 1.3 :Concerns to Wider Adoption of Cloud (Cisco,2012)

There are many of encryption algorithms that were used in Cloud Computing to protect data and provide the type of privacy, such as, Amazon EC2, AES-256, CipherCloud, a Cupertino, Calif.-based start-up and others. Also, many of algorithms used to improve the security and privacy within the Cloud Computing but still there are some concerns and aspects untrusted.

This thesis will study some of the models or systems proposed recently in the field of data encryption Cloud Computing, in order to benefit from these researches to build our model.

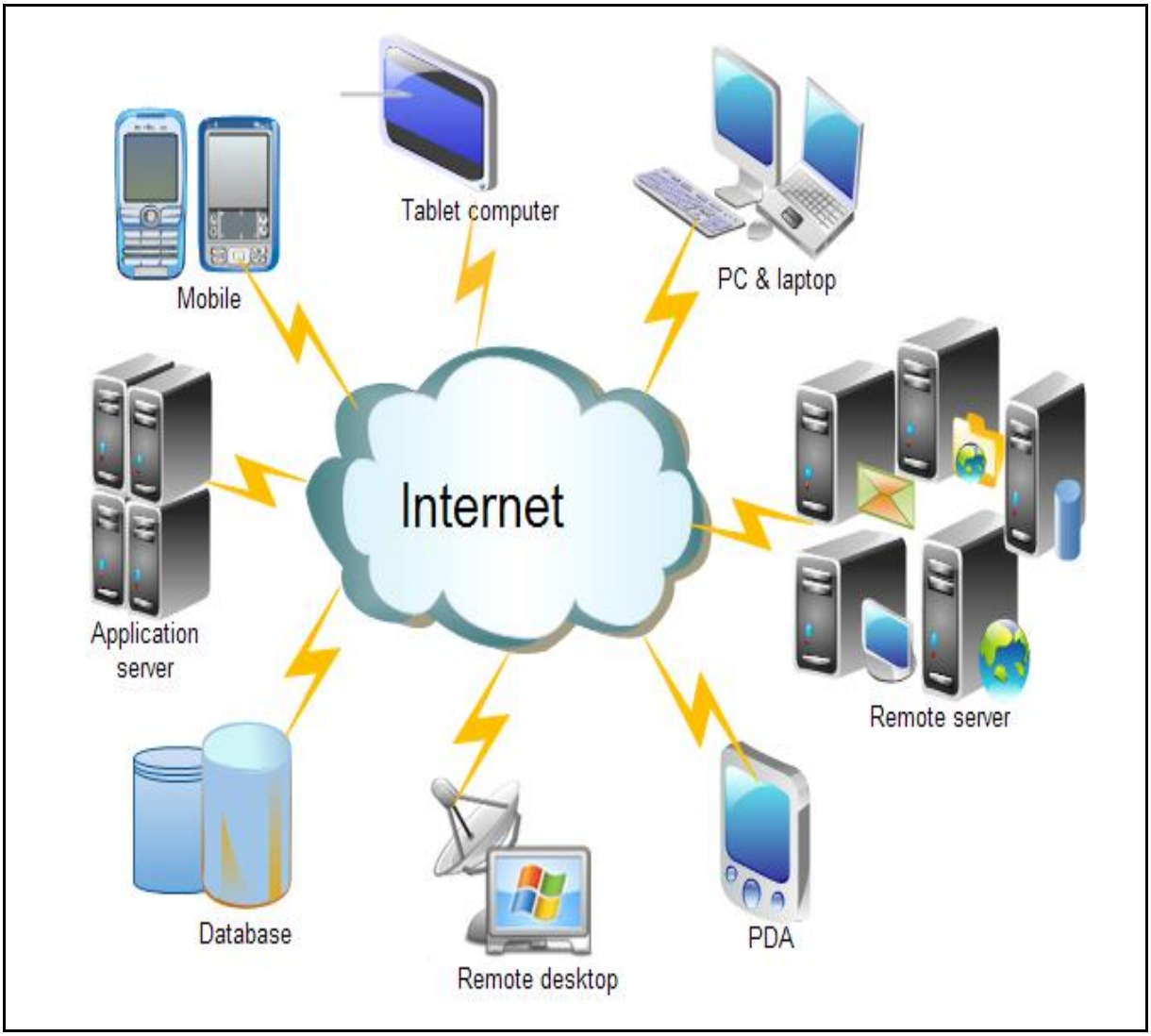

Figure 1.4: The Cloud Computing Environment

1.2 Problem Definition:

The recent developments in Cloud Computing allows for organizations and users to use different applications and store their data on a Cloud.

Through the study and research in the process of storing the encrypted data within the Cloud .We observed many problems, these are:

1- All types of data are stored using the same encryption algorithms.

2- The cost of storing the data on Cloud is high. Here will need more space for storage.

3- The required time to encrypt and decrypt the data to/from the Cloud is long.

All the mentioned problems are because of there is not clear method to split the data into various classifications or levels to make each level uses different encryption algorithms depending on the degree of important.

1.3 Study Objectives:

The main objectives can be summarized as the following :

1 - Encrypt the data to be stored in Cloud according to their importance.

2- Reduce the time of data encryption, as well as, decrease the cost of storage and retrieval of data stored in the Cloud.

3- The disparity in data encryption which makes data security within the Cloud is varies, therefore, make data security powerful and very difficult against the intrusion and hacker operations.

4- Encrypt each file by different encryption key(s), which aims to increase the protection of privacy and prevent its violation by the hacker, or even by the Cloud Computing service providers themselves.

5- Protect the data as much as possible, to make the losses or the penetration of data in

case of occurrence is very limited.

6 - Store the data in different Cloud providers depending on the level of importance.

1.4 Significance of the Study:

Importance of this study lies in the ability of the proposed method to reduce the cost of computation and storage and improve the security in comparison with the existence methods. The proposed technique reduces the time required to encrypt and decrypt the data. Furthermore, the time required to send and receive that data will be reduced as result of using different encryption algorithms.

1.5 Limitations of the Study:

- A completely secured system stills a very difficult; because the growth of information systems networks and their infrastructure led to growing in attack methods and rates against information systems environment.
- The accuracy of protection cannot be achieved as 100%.
- The encryption algorithms that used are not new, to accomplish a new encryption algorithm; it needs long-time to be implemented and validated.
- Using more than one encryption algorithm to protect the data can lead to more overhead. Therefore to implement this approach it requires using very effective way.

1.6 Thesis Organization:

This thesis consists of five chapters including this chapter which presents an introduction to thesis, problem statement, also it gives the objectives of research, discusses the significance and limitation of the study and finally it presents thesis organization. Chapter two reviews the Cloud Computing approaches, also presents an

overview about Encryption Algorithms, gives an overview the categories of attacks, and finally it lists the related work. Chapter three introduces the research methodology used in this thesis, as well as, discusses the data partitioning method, presents the algorithms that used in the encryption process and finally illustrating the software that has been used for evaluation the model. Chapter four illustrates detailed experiments about our model, presents the experimental results that obtained from our model, and finally a comparison with other studies results is made. Finally, chapter 5 concludes the entire study and then presents conclusions and recommendation for future work.

Chapter Two

Cloud Computing, Literature Review and Related Work

Chapter Two

Cloud Computing, Literature Review and Related Work

2.1 Introduction.

This chapter consists of five sections. Section 2.2 discusses the Cloud Computing approaches; section 2.3 gives an overview about Encryption Algorithms; section 2.4 discusses the categories of attacks; and, Finally section 2.5 lists the works related to this thesis.

2.2 Cloud Computing Approaches.

Plummer et al. (2008) defined the Cloud Computing as a model of delivering Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) in which the customer pays to use, rather than own, computational resources. The main goal of Cloud Computing system is to shared service delivery. According to (NIST, 2011) , there are two main types of Cloud Computing Models: service model and deployment model.

2.2.1 Cloud Computing Architecture

As shown in Figure 2.1, The Cloud Computing Architecture and divided into four components:

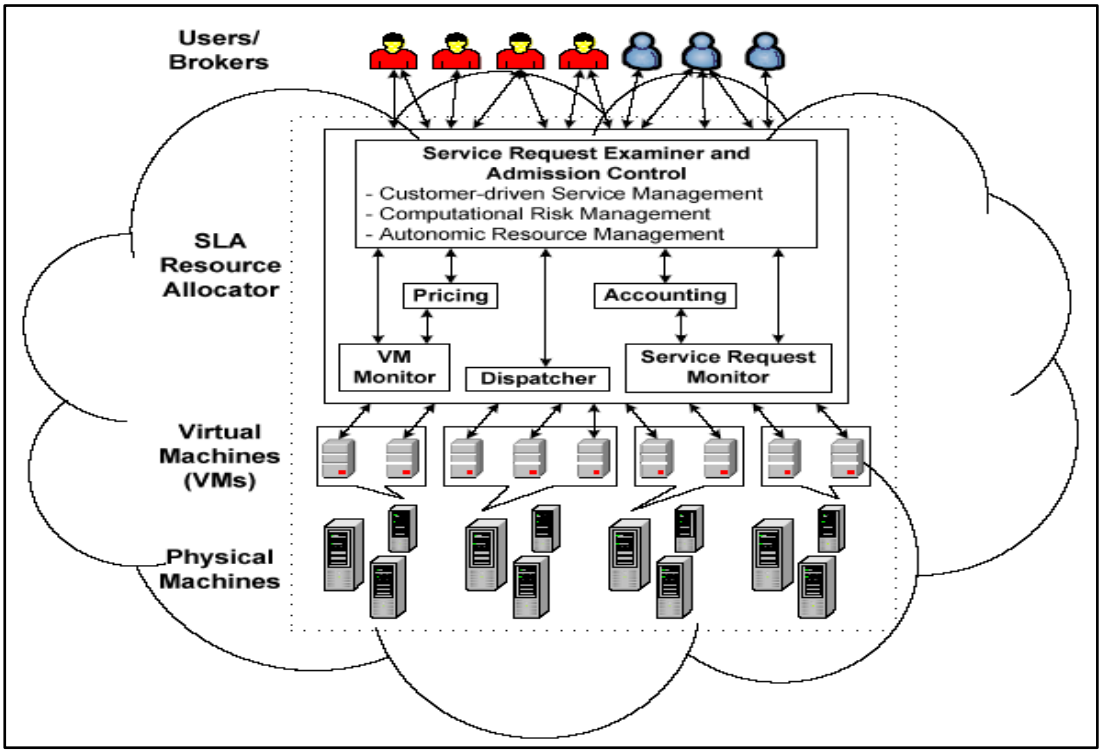

Figure 2.1:The Cloud Computing Architecture and Components (Buyya, et al., 2009).

- **Brokers or Users**: broker works to present service requests to data centers and Cloud on his behalf so that to be processed.
- **Service Level Agreement (SLA) Resource Allocator**: used to providing the interface between external users (brokers) and the Data Center in Cloud service provider.
- **Virtual Machines (VMs):** the most important component in the Cloud are VMs. Can be created many VM scan in a single machine to meet accepted service requests, this is necessary to provide maximum flexibility in establishing various partitions of resources on the same machine with different requirements of services.
- **Physical Machines**: are huge computing servers and multiple within the data centers, they are using for provides resources to meet service demands.

2.2.2 Cloud Computing Service Models.

It consist of three layers which represent the three main types of Cloud service as shown in Figure 2.2

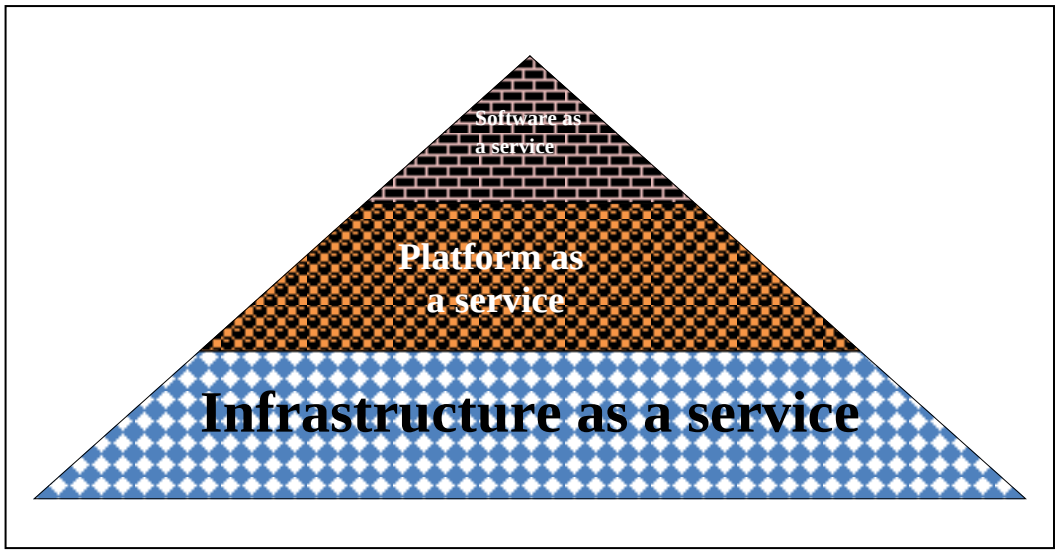

Figure 2.2: The Cloud Computing Service Models

- **Software as a service(SaaS) :** It represents the top layer , and the main idea is to given the users or consumers the ability to access and use variety applications or services that is hosted in the Cloud through a centralized network that can access over the internet or intranet (Hassan , 2012). The examples for software as a service include Google Apps, DeskAway , Wipro w-SaaS , internet email, Google docs and salesforce.com (Madhavi , et al., 2012).
- **Platform as a service (PaaS):** It represents the middle layer, the main idea is enables the users or consumers for the design and deploy of their own applications (softwares) through a development platform (Ding, et al., 2012). The resources physical and logical such as the operating systems and network access are not managed by the users. The examples for platform as a service

include Microsoft Windows Azure, Google App Engine, OrangeScape and Wolf PaaS (Zou, et al., 2012).

- **Infrastructure as a service(IaaS) :** Finally, the base layer which enables users or consumers for control and manage the processing, storage, and network connectivity , but it does not allow them to control the Cloud infrastructure (Madhavi , et al., 2012). The examples for infrastructure as a service include Amazon EC2 (Elastic Compute Cloud), Amazon S3 (Simple Storage Service), Google and IBM (Jadeja and Modi, 2012).

2.2.3 Deployment Models.

Depending on requirements or needs for companies or consumers to deploying the Cloud Computing. There are four main deployment models in Cloud, as shown in Figure 2.3, each of them has its own features that support the needs of the Cloud users of services in different ways (Kulkarni , et al., 2012).

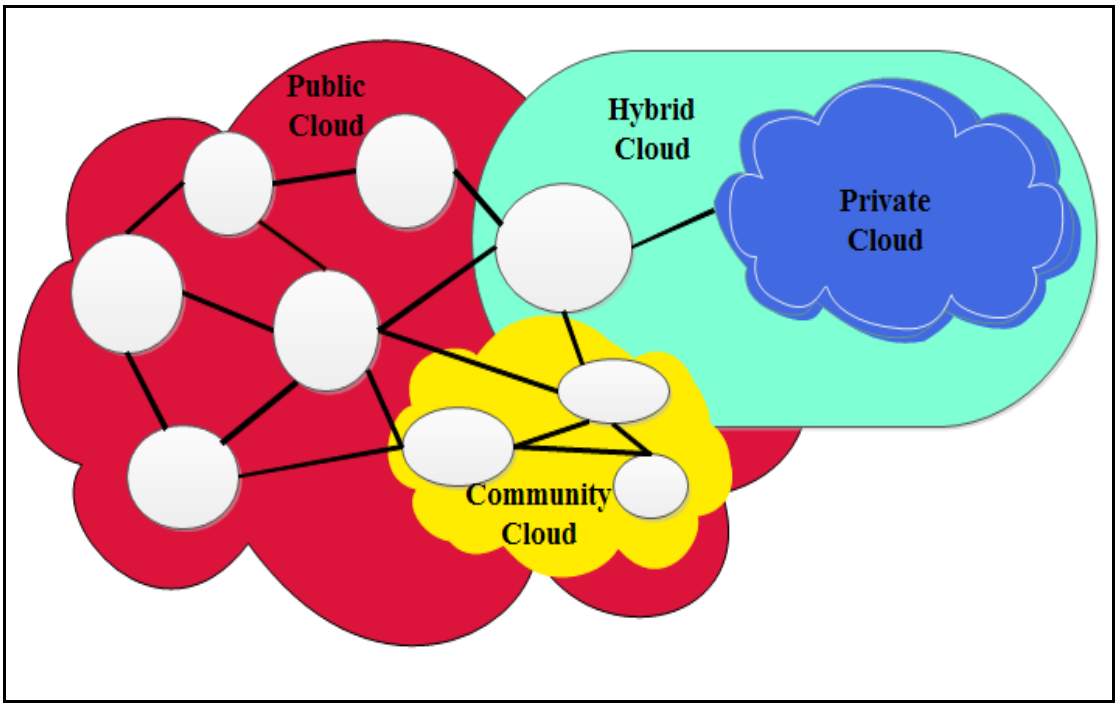

Figure 2.3 : The Cloud Computing Types

- **Private Cloud:** Also called Internal Cloud or Corporate Cloud, this type is established for one organization(or specific group), the infrastructure is managed and operated by this organization or a third party (Qaisar , et al., 2012). The purpose to keep a consistent level of privacy, security, and access control via an Application Programming Interface (API). It can may exist on premise or off premise (Jadeja and Modi, 2012).
- **Public Cloud:** Also called External Cloud, in this type , the Cloud is established for the open use by the general public ,the infrastructure is managed and operated by a business, academic, or government organization, or some combination of them (Qaisar , et al., 2012). It can may exist on premise or off premise.
- **Community Cloud:** This type of Cloud is established for several organizations and supports a specific community that have similar requirements and sharing the same filed , the infrastructure is managed and operated by this organizations or a third party and can exist on premise or off premise (Savu ,2011). Example is the Open Cirrus Cloud Computing Testbed.
- **Hybrid Cloud:** The last type of Cloud combines number of Clouds of any type (private, community, or public), at least it combines two Clouds of any type, The main prerequisite for blending or combines of this type of Cloud is a consensus among the Clouds allowed the transfer of data or applications from one Cloud to another (Mazhelis and Tyrväinen, 2011). The infrastructure is managed and operated some resources by this organization in-house and has others managed externally (Qaisar , et al., 2012).

Finally, CloudTweaks (2012) discussed some benefits and drawbacks of Cloud Computing types in its published article as shown in Table 2.1.

Table 2.1: Benefits and Drawbacks of Cloud Types (CloudTweaks, 2012).

	Benefits	Drawbacks
Private Cloud	higher level of security and regulatory compliance, higher levels of performance because virtual servers that don't need to share data buses or processor time with the "noisy" servers of other companies, more cost-effective datacenter management and higher levels of scalability. And higher level of control over your data. avoid "vendor locks"	higher initial investment, less flexibility , higher cost of maintenance and more complicated to setup and more expensive to maintain a data security requirement.
Public Cloud	Cost: organizations can trim their IT budgets (physical hardware - energy costs - virtual hosted at a third party - specific storage parameters, applications, and monitor the system, and security options).	Lack of Control Due to the fact that third party providers are in charge of storing and maintaining the data systems, the Speed Due to it based on internet connections, meaning the data transfer rate is limited to that of the Internet Service Provider (ISP). Also Lack of Investment.
Community Cloud	-The Cloud expertise is developed within the community. -The riskiness of working with one Cloud vendor is mitigated. -The solutions are developed to overcome problems of the community. -The efforts are consolidated across community which produces a robust and diverse service and platform.	A considerable coordination is required among Cloud community members to agree upon the Cloud provider and central location issues.
Hybrid Cloud	Combine the advantages of public Cloud and private, also comes up with an ideal approach for local infrastructure with scalable infrastructure that is provisioned on demand, also offers the security, the substantial cost savings, and maintaining control on private Clouds. The hybrid model enabling unlimited flexibility.	Special expertise is required to integrate the public and private Clouds which make it the most complex Cloud solution to manage.

2.3 Encryption Algorithms Overview

Any type of deployed Cloud depends on the security protection needed for the

data involved**.** Therefore, there are many ways used to achieving this goal, such as Encryption. Encryption is the process of finding appropriate ways to convert a clear information (plaintext) to unclear texts (Ciphertext), to prevent unauthorized persons have access to it (Pan,2011) **.** Many encryption algorithms have been suggested and implemented throughout the ages by potential users of this type of science. Using encryption to achieve many objectives, such as Confidentiality, Integrity, Authentication and Non-repudiation.

Encryption can be classified into two main types Symmetric and Asymmetric cipher models .Table 2.2 shows basic terminology of encryption.

Table 2.2: Basic Terminology of Encryption (Piper, 1996)

Terminology	Description
Plaintext	Clear text (original message) to be encrypted
Ciphertext	Unclear text (the encrypted message)
Enciphering (encryption)	The process of transforming plaintext into ciphertext
Encryption algorithm	Executes encryption by two inputs (plaintext and secret key).
Deciphering (decryption)	The process of retrieving plaintext from ciphertext
Decryption algorithm	Executes decryption by two inputs (ciphertext and secret key).
Secret key	One key (symmetric key) used for encryption and decryption
Cipher (cryptographic system)	A scheme for encryption and decryption
Cryptography	Science of studying cryptographic system
Cryptanalysis	Knowledge of studying attacks against ciphers
Cryptology	Combining cryptography and cryptanalysis
Symmetric cipher	The encryption and decryption are using the same key(one key)
Block cipher	Every time encrypts a block of plaintext data (usually 64 or 128 bits)
Stream cipher	Every time encrypts one bit or one byte of plaintext data (usually 1 bit or 1 byte)
Asymmetric cipher	The encryption and decryption are using the different keys (at least two keys)

2.3.1 Symmetric Cipher Model

Symmetric Cipher Model also referred as (secret key, conventional or unique key encryption). The encryption / decryption process uses the same key by the same encryption algorithm as shown in Figure 2.4. Used widespread algorithms in this type such as AES (Rijndael) , Serpent, Twofish , Blowfish , CAST5, RC4, DES , 3DES, and IDEA (Breveglieri, et al., 2007)

Symmetric cipher use two types of ciphers:

- **Stream ciphers:** usually it encrypts one bit or one byte of plaintext data (1 or 8 bits).
- **Block ciphers:** usually it encrypts a block of plaintext data (64 or 128 bits).

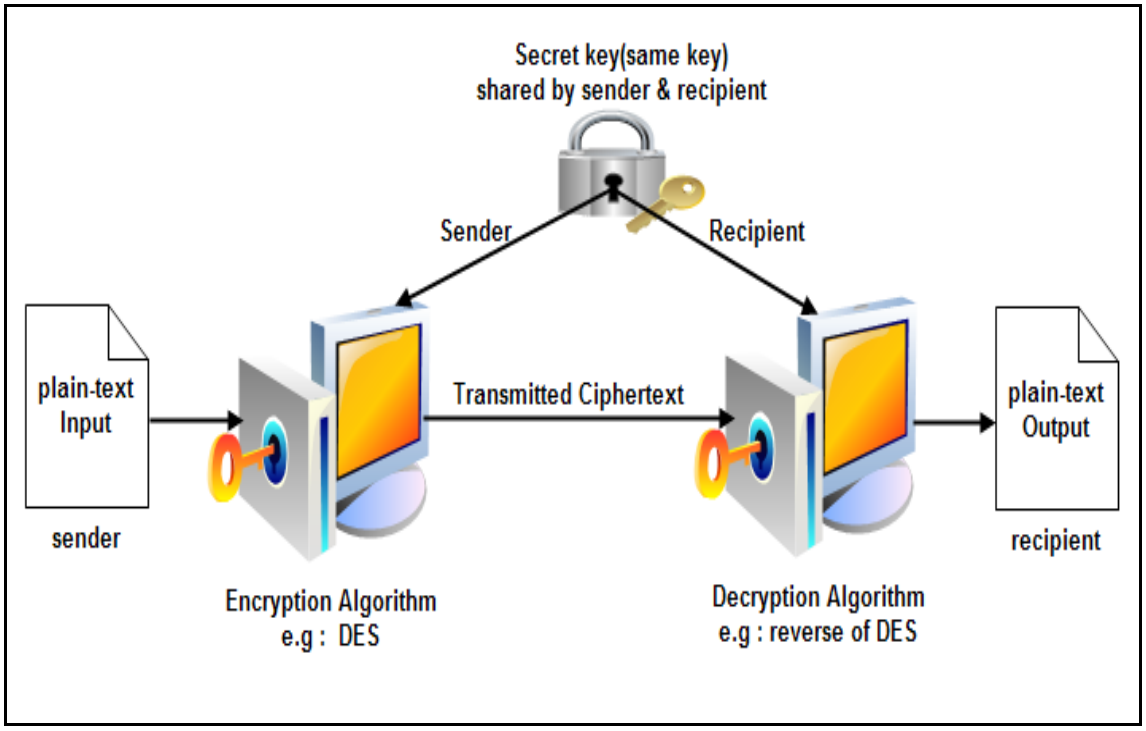

Figure 2.4 : The Architecture of Symmetric Cipher

2.3.2 Asymmetric Cipher Model

Asymmetric Cipher Model, also referred as public-key cryptography, it uses

two related keys (public and private) , If it is uses the public key to encrypt any message then only it be decrypted by the private key of the same algorithm , Also, If is using the private key to encrypt any message then only be decrypted by the public key by the same algorithm as shown in Figure 2.5. Asymmetric algorithms the most common used such as RSA, DSA, ELGAMAL, and ECC (Menezes , Oorschot and Vanstone, 1996) .

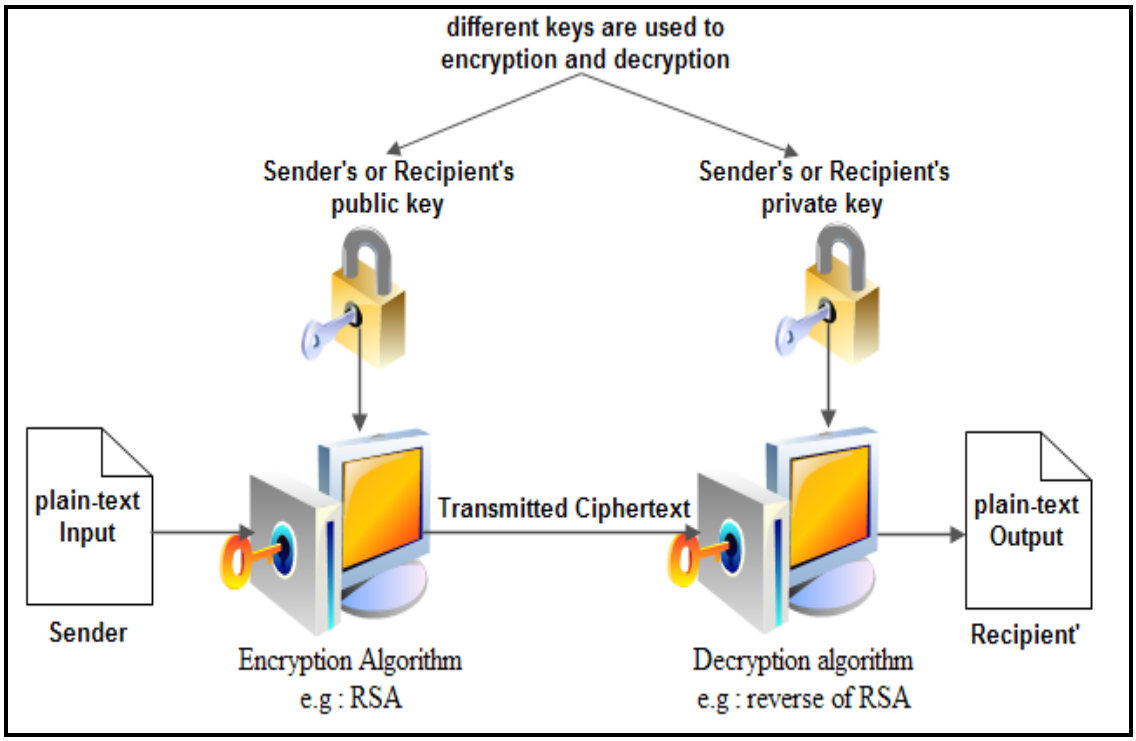

Figure 2.5 : The Architecture of Asymmetric Cipher

2.4 Attacks Overview

(Singh & Shrivastava 2012) have collected attacks types and categorized them in five categories. These categories are: Denial of Service (DOS), Cloud Malware Injection Attack, Side Channel Attacks, Authentication Attacks and Man-In-The-Middle Cryptographic Attacks as shown below.

2.4.1 Denial of Service (DOS) Attacks

In this type of attacks, the attacker tries to prevent authorized users from

accessing to the service or use. The server becomes unable to respond to the regular consumers due to sending it in short time thousands of requests from the attackers (Bhadauria & Sanyal 2012). These attacks are Ping of death, Teardrop, Mailbomb, Smurf, Land, Apache2, SYN Flood, (MIT, 1999).

2.4.2 Malware Injection Attacks

The purpose of the Malware (malicious software) Injection Attacks is to inject a malicious virtual machine or service implementation into the Cloud system which could be used later to serve any particular objective the adversary is required to do (Singh & Shrivastava 2012). In this type of attacks, the adversary implement its own malicious module and add it to the Cloud in a way that make the system treats this malicious module as a valid instance for the particular. If this succeeds, the malicious module code will be executed when the system redirects automatically valid user requests to the malicious service implementation. The main idea of this attack is that an adversary uploads a treated copy of a service instance of the victim such that some service requests to the service of victim are processed within that malicious module.

2.4.3 Side Channel Attacks

In this type of attack, the intruder place malicious virtual machine in close proximity to a target Cloud server to attack and then launching a side channel attack. The side Channel Attack exploits the side information that could be retrieved from the encryption device such as power consumption statistics or timing information (Singh & Shrivastava 2012). These information could be used along with crypto analytic methods to get the key of the device.

2.4.4 Authentication Attacks

Authentication is follow procedure specific to prove and verify the identity of the user or entity who (he/she) claims to be in Cloud. The authentication is very important and critical because all data (specially the sensitive) stores at Cloud service providers. This indicates that the attacker can work as a service provider for hosting the data and obtains all security information and passwords about the stored data**.** Transactions, electronic business and other activities that depend on internet need a more stringent authentication process**.** Passwords, digital certificates (DC), Digital Signatures (DS), finger print, and audio. In addition, picture is considered as a solutions and procedures to perform authentication on the Internet.

2.4.5 Man-In-The-Middle Cryptographic Attacks

In this type of attacks, the attackers place themselves secretly in the communication's path between two parties (two Clouds). The target to control on communications through intercept or modify unencrypted the information. This type of attack will occur if secure socket layer (SSL) is not exactly configured (Qaisar , et al., 2012).

2.5 Related Work

The rapid development of the Cloud Computing encouraged many of researchers and companies to focus and research in all aspects of the Cloud in general, but a many of researches were focus on the area of data protection and privacy. There are many proposed researches to make data more secure, also give the Cloud's users an effective of high privacy by many different methods and using different techniques. This section will look at some of these researches , and the

following is brief of some related works:

Shen, Shi & Waters (2009) presented a new encryption approach and they named it "searchable encryption". This approach allows the owner data to be encrypted by any encryption algorithm, and then he can authorize another party to conduct searches on a pre-specified set of keywords without giving the third party any powers to disclose any sensitive information. The lacks of flexibility is one of the disadvantages associated with this approach especially when trying to obtain data in addition to one-place storage which let the data exposed for threats and danger.

Yuefa et al. (2009) have designed Data Security Model for Cloud Computing. This model uses three-level or three-layer system structure. The first layer is responsible for user authentication and the second layer is responsible for user's data encryption and user's privacy protection using a specific and efficient method. Finally, the third layer is responsible for the fast recovery of the user data. Within the encryption layer, they used two keys; one for public encryption and another key for privacy protection.

Chen & He (2010) have suggested a new strategy for the storage of data security. This strategy depended on the principles relating to reparation and the theory of the initial setup. Purpose of this strategy is to increase the number of data blocks. Proceeded this strategy to divide the data blocks to prepare more and encrypt each block by different encryption key. Splitting data into blocks is implemented by an algorithm called data partitioning algorithm. The benefit of increasing the number of blocks is to increase the security of data within the Cloud. Instead of full data encryption in a limited number of keys turned an innovative strategy to split the data into blocks in order to increase the numbers of encryption keys. But it has shortcomings, for example the division process the data blocks for increasing the

number of blocks will lead to data redundancy significantly and also will increase the number of encryption keys , the greater the number of blocks will increase the number of encryption keys and this increases the cost of storage. Also there is a problem when there are repeated blocks containing the same data, every one of them need a different encryption key.

Puttaswamy, Kruegel & Zhao (2010) presented an approach or study to encrypt sensitive data and then storing the encrypted data at a third party. This non-trusted third party only will be an intermediary between the owner of the data and the users of the data. The third party does not know the keys to decoding so it cannot penetrate the security of data or intruding on the privacy of the data. Performing operations on data carried out by programs installed for the owners of data with trusted users, then stored in a third party. To decrypt the data and processing only by the owners of the data or by the customers trusted. What noted here is the data processing carried out in various places, so we need to program in more than one place, and this increases the cost (such as cost of hardware , software and required storage). The more data users the greater the cost. Also we noted that there is a lack of access to the full features of Cloud Computing. Only we benefited from the process of storage.

Atayero & Feyisetan(2011) proposed a new encryption layer contain or encapsulate all encrypted files before storing them in the Clouds. This extra layer would be an encrypted search index layer .After the files are encapsulated in this layer; files are stored within the Cloud. The main purpose of this layer is to increase data security and privacy. They used a symmetric encryption method to make this layer works as a search index layer. Each file has its own index such that we can use this security index to look for the file within the Cloud. One of the advantages of

Searchable Symmetric Encryption is that it permits a user to selectively search the data that the user hosted in the Cloud.

NEXSAN (2011) Assureon is design Assureon archive storage systems that are use separate AES-256 encryption for each Cloud services customer. Additionally, each file is individually encrypted with its own AES-256 key to provide the strongest separation and security of data for each client. Assureon is designed for providing an online archive for high value content with a high performance and secure.

Pagano D & Pagano F (2011) this proposal is suitable in the case of distributed databases with enormous size. Authors presented a solution to encrypt data. This solution was adopted to encrypt the data on each row level “encryption each row of tables that are stored data by different encryption key." This solution was allowed to increase confidence in the protection of privacy sensitive data shared between users and distributed data in different places. The main advantage of this technique is the ability to define access control to a subset of data (rows) of the table based on the distribution of decryption keys. Although this solution has increased the level of privacy not to be observed that he focused on the side of the distributed data encryption. In the case of large distributed databases, for example, the U.S. health insurance database, encryption process at the level of row size this means many numbers of encryption keys this will increase the size of data storage and thereby increasing the cost.

Sarode, Giri & Chopde (2011) proposed the user interface model. Service provided by the model is described in the three Cloud systems, the first is the encryption/decryption system, the second is the storage system and the third is the user interface implementation system. What concerns us in this model is to

understand the system of encryption and decryption. The system of encryption and decryption has different ways to ensure that the most important one when we want encryption or decryption verifies the credibility of the traffic only once this is done using an algorithm RSA, which operates under the public key and private key. To clarify when a user wants access to certain data will be as follows:

1 - Allows the user to login once by checking the credibility of the traffic to the model and the user interface to get to the data.

2 - The user sends a request to obtain the data found within the Cloud through the user interface model.

3 - Data is sent encrypted from storage Cloud service to the encryption / decryption service.

4 - The encryption / decryption service handles data and sends it as a service to user interface model.

5 - Data is sent to the user.

Porticor (2012) introduced system; This system combines cryptographic service and management of keys encryption and keys decryption. He combined them together to protect the data inside Cloud. This system uses a unique key which is divided into two keys, one is called public key and the second is called the private key. The public key encrypts all data blocks involved in the application or a single program, " this exactly means that all data blocks of a program are using the same public key " While the private key encrypts each block of data alone, ""i.e., each block of the same program it uses by the private key is different from the other." Note that the public key cannot know any information about the private key.

Here note that this idea of protecting the data by this system it's strong. But note that this system did not provide a clear idea of how to protect data privacy,

especially as the supplier of the work of the private keys is the same system offering users service work them keys own.

Kumbhar , Chaudhari & Badhe(2012) proposed a business model for Cloud Computing. This model is built on the separation of encryption service and between decryption service about storage service provided by each service provider Cloud which makes each service operates its work away from the other service. The purpose of the process of separating encryption service and decryption service away about the storage area is to make both of them in a different place to increase data security. This model was divided into three phases. The first phase for encryption and the second phase for the separate storage and the third phase for decryption separate. This model was used in the process of encryption RSA algorithm.

The process of separating from the other, each service has a high cost, where cost is divided into more than one. Also when retrieving data from the owners will take more time. There are also risks in the event of a disruption or loss in the decryption service this will lead to an inability to work on encrypted data in another place.

Chapter Three

Methodology and Proposed Model

Chapter Three
Methodology and Proposed Model

3.1 Introduction.

In general, the research methodology consists of the problem, collecting the data or facts, analyzing the data and reaching certain conclusions in the form of solution(s) towards the concerned problem. In this chapter, we present a detailed description of the proposed encryption technique, as well as, discuss the data partitioning method, and finally illustrating the algorithms that used in the encryption process.

3.2 The Proposed Model Architecture

The proposed model architecture consists of three levels, as shown in Figure 3.1.

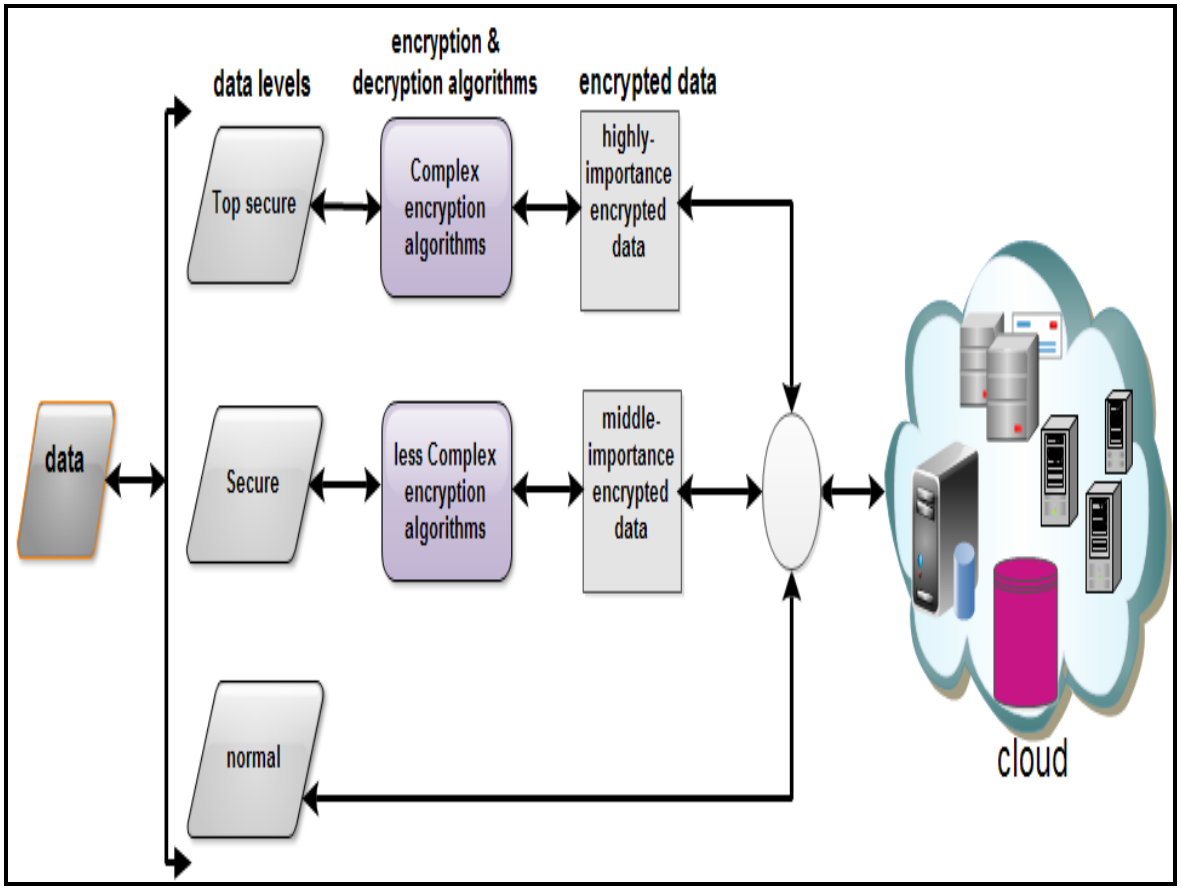

Figure 3.1: The Architecture of The Proposed Model

In the following subsection we will give a detailed description.

3.2.1 Data Levels

This phase takes the data and divides it using the Data-Segmentation Algorithm (as described in details in the next page) into three levels according to its importance , these levels are as follows:

Level 1 : Top Secure-Data

The data in this level is classified by the data owner as the most important and most sensitive data depending on set of specific measures. The data owner is responsible for determine these measures.

Level 2 : Secure-Data

The data in this level is classified by the data owner as middle important depending on set of specific measures. The data owner is responsible for determine these measures.

Level 3 : Classified-Data

The data in this level is classified by the data owner as a regular data that need no encryption at all. Classification of data in this level based on set of specific measures. These measures depend on the data owner point of view.

3.2.1.1 Data-Segmentation Algorithm (DSA)

There are many ways to partition the huge data into set of proper and manageable size, such as horizontal or vertical segmentation, hybrid (horizontal and vertical), and database segmentation into subsets depending on specific criteria.

Data varies from one field to another, so the data owner is responsible for data classification to appropriate subsets according to their importance, depending on specific measures, such as the following:

1- The degree of security and protection required for each dataset or group (which will be used in this theses interchangeably).

2 - The required size for each group.

3 - The size of data that generated after encryption.

4 - The time required to store /retrieves each dataset.

5 - Create index file containing information about encrypted and stored dataset on Cloud.

Segmentation the data into groups (as shown in Figure 3.2) helps in reduce the storage and encryption costs, also increases the communication speed between the data owner and the cloud's services provider. For the hybrid segmentation (horizontal or vertical) and, also for database segmentation we proposed two methods for grouping data according to their importance level.

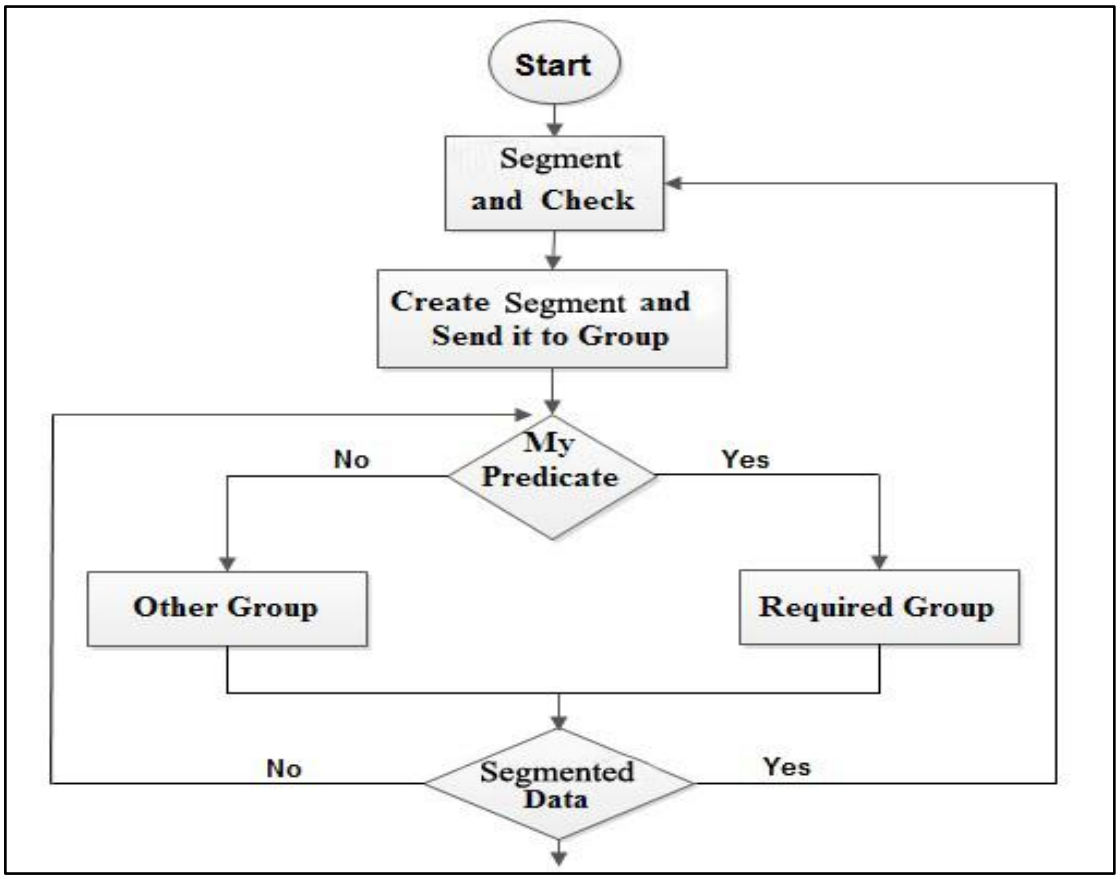

Figure 3.2 : The steps of segmentation algorithm

Following is the description of the algorithm that executes the hybrid segmentation :

The Hybrid Segmentation Algorithm (HSA):

Input:

1- relation *R ; R* consists of rows *I* and columns *J.*

2- set of predicates Pr = {p_1, p_2, …,p_k} For relation *R*

we define p_i : A_j θ V

where $\theta \in \{ >,\geq,<,\leq,\neq,= \}$, V $\in$ D_j and D_j is the domain of A_j .

3- Pr = {p_1,p_2} which is complete and minimal *Pr = Pr'* = {p_1,p_2,…,p_k}

Output:

Set of segments of *SE* = {SE_1, SE_2,…,SE_n } which accept the segmentation conditions.

Where R= $\bigcup_{\forall Ri \in SE} Ri$

Begin

Repeat

For I = 1 to the number of rows in the relation R

For J = 1 to the number of columns in the relation R

If I $\in$ Pr and J $\in$ the same Pr then

Rows I and (Columns J or part of Columns J) are grouped together in the same segment; set 1 to the segment entry, and increase I by 1.

Else

Rows I and (Columns J or part of Columns J) are grouped in different segments ;

set 0 to the segment entry, and increase I by 1.

End if

End for

End for

Until all relations in the database have been processed

End.

Note: to get the vertical segmentation, we will use the same above algorithm with changing the inner loop with the outer loop.

To illustrate the segmentation approach, we assume database with a specific size. We segmented it into ***n*** segments from $SE_1 \ldots SE_n$ depending on the degree of importance and required protection level as shown in Figure 3.3. Also, we assume exist different three levels from L_1 to L_3 to encryption and storage these segments .We generates ***m*** groups from G_1 to G_m and their respective segments as shown in table 3.1.

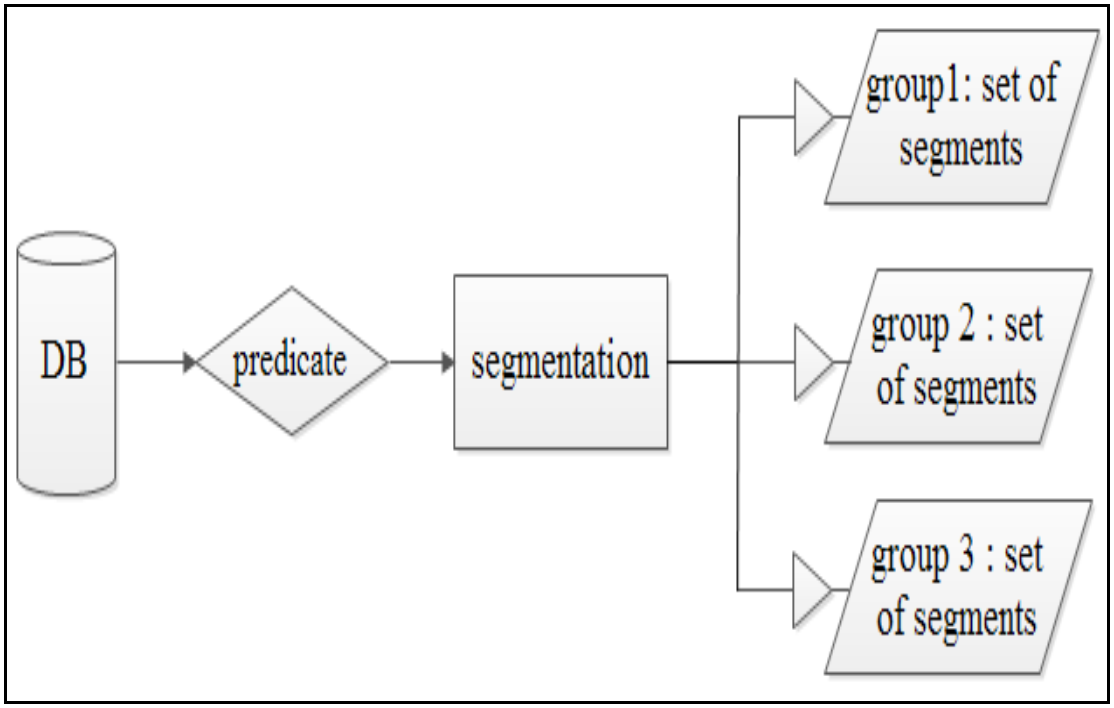

Figure 3.3 : Segmentation the Data and Grouped into Different Levels

$\mathbf{SE_i}$ is each segment.

$\mathbf{L_j}$ is each level.

$\mathbf{G_n}$ is each group.

Table 3.1 : Generated Groups and Their Segments

Group#	Segment#						
	SE1	SE2	SE3	SE4	SE5	SE6	SE7
G1	1	0	0	1	0	0	0
G2	0	1	0	0	1	1	0
G3	0	0	1	0	0	0	1

Following is the description of the algorithm that executes the database segmentation:

The Database Segmentation Algorithm(DSA):

Input:

1- Database **DB; DB** contains set of files $\mathbf{F=\{F_1, F_2, \ldots, F_n\}}$.

2- Set of predicates $\mathbf{Pr = \{p_1, p_2, \ldots, p_k\}}$ For database **DB**

we define $\mathbf{p_i : F_j} \ \boldsymbol{\theta}\ \mathbf{V}$

Where $\boldsymbol{\theta} \in \{\neq, =\}$, $V \in D_j$ and D_j is the domain of $\mathbf{F_j}$ (such as index or name).

Output:

1- Set of segments of $\mathbf{SDB = \{SDB_1, SDB_2 \ldots SDB_n\}}$ which accept the segmentation conditions.

Where $\mathbf{SDB_j}$ contains set of files $\mathbf{F =\{ F_1, F_2, \ldots, F_n\}}$ and $\mathbf{SDB_j}$ different about $\mathbf{SDB_{j+1}}$

2- If database DB segmented into set of $\mathbf{SDB = \{ SDB_1, SDB_2, \ldots, SDB_n\}}$ then

$\mathbf{DB}=\bigcup_{\forall \mathbf{DBi} \in \mathbf{SDB}} DBi$

Begin

Repeat

For I=1 to ***n*** , where ***n*** is the number of files in the database DB

If Pr = V , where V either file name or file number

then

Files that ∈ Pr are grouped together in the same segment; set 1 to the segment entry, and increase I by 1.

Else

Files are grouped in different segments; set 0 to the segment entry, increase I by 1.

End if

End for

Until all Files in the database have been processed

End.

To illustrate the segmentation approach ,we assume database with set of ***n*** files from **f1…fn** . They are different in the degree of importance and required protection level as shown in Figure 3.4. Also, we assume existing different three levels from $\mathbf{L_1…L_3}$ to encryption and storage these files. We generate ***m*** groups from G_1 to G_m and their respective files as shown in table 3.2.

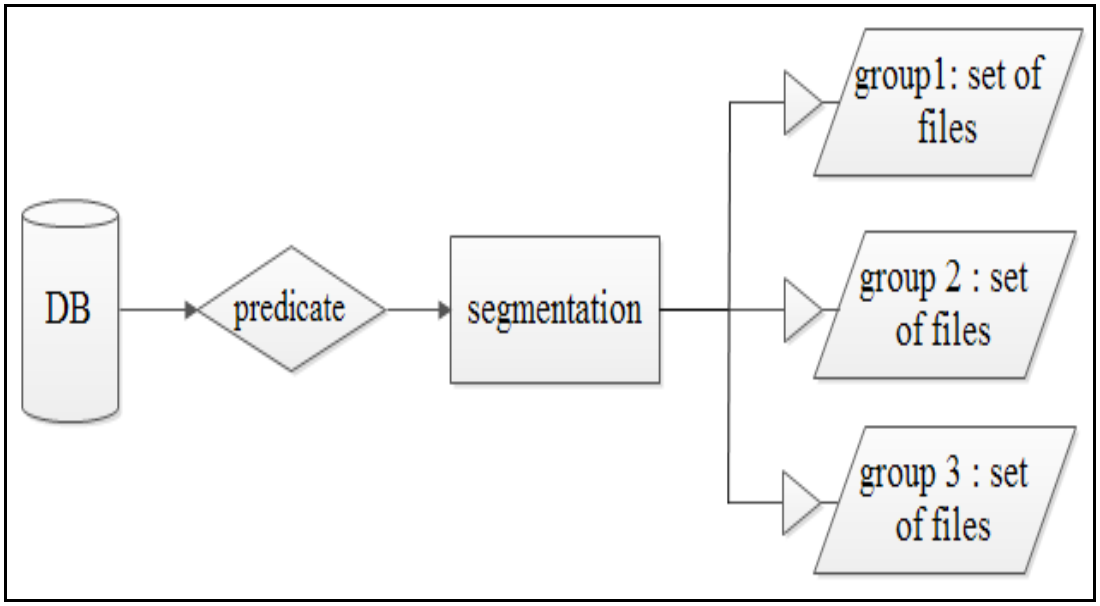

Figure 3.4 : Database Segmented to Files and Grouped into Different Levels

$\mathbf{F_i}$ is each file.

$\mathbf{L_j}$ is each level.

$\mathbf{G_n}$ is each group.

Table 3.2 : Generated Groups and Their Files

Group#	File#						
	F1	F2	F3	F4	F5	F6	F7
G1	1	0	0	1	0	0	0
G2	0	1	0	0	1	1	0
G3	0	0	1	0	0	0	1

The proposed algorithm for allocating each group into the appropriate level according to their importance as shown in table 3.3. The following is the description of the proposed algorithm.

Input:

1- set of levels $L=\{L_1,L_2,\ldots,L_K\}$ For database DB

we define $L : L_i \ \theta \ V$

where $\theta \in \{\neq,=\}$, $V \in$ { Top-level , Middle-level , Down-level} and V is the domain of L_i .

2- set of groups $G = \{G_1,G_2,\ldots,G_n\}$ where each G_j contains

either set of segments $\boldsymbol{SE = \{SE_1, SE_2,\ldots,SE_n\}}$

or set of files $\mathbf{F=\{f_1, f_2, \ldots,f_n\}.}$

Output:

set of levels $L=\{L_1,L_2,\ldots,L_K\}$

Where each L_i contains one group or more of $G = \{G_1, G2,\ldots,G_n\}$.

Begin

Repeat

For I = 1 to number of Levels proposed L_K

value V_1= Top-level

value V_2= Middle-level

value V_3= Down-level

For J = 1 to number of groups in database G_n

Assign group Gj to level Li depending on owner measures

Table 3.3 : The Levels with Their Groups

Level#	Group#		
	G1	G2	G3
L1	1	0	0
L2	0	1	0
L3	0	0	1

3.2.2 Encryption Algorithms

There are many encryption algorithms used to protect the data , some of these algorithms: Data Encryption Standard (DES) , Triple Data Encryption Standard (3DES), Advanced Encryption Standard(AES) , Elliptic Curve Cryptography (ECC) , Blowfish and other (Umaparvathi and Varughese, 2010). Also, when talking about any encryption algorithms it means these are algorithms same as used to decrypt the encrypted data. High-complexity encryption algorithms will used to protect the high important data and less-complexity encryption algorithms will be used to protect the data with medium importance.

1-High-Complexity Encryption Algorithms

The complex algorithms are used for encryption the high-importance data; there are many algorithms such as, the Elliptic Curve Cryptography (ECC), RSA algorithm,

Advanced_Encryption Standard algorithm (AES) or Triple Data Encryption Standard algorithm (3DES). The data owner can use one or more algorithm according to his needs. Also each file is encrypted by a key or more different from the rest of the files in the same level.

As an example we can choose the ECC algorithm , which will be described below:

The Elliptic Curve Cryptography Algorithm

The ECC is type of Asymmetric Cryptography (or, Public Key Cipher) in 1985 discovered as a formal standard for encryption by Victor Miller (IBM) and Neil Koblitz (dKrypt,2013). ECC can only encrypt and decrypt a set of points on the curve Ep(a, b), set of solutions (x,y) to an equation of the form $y^2 = (x + ax + b) \bmod p$ (p is prime number), where $(4a^3 + 27b^2) \bmod p \neq 0$ (Tawalbeh, Mowafi and Aljoby, 2012). This Means not encrypt/decrypt messages. Recently , ECC is one of the strongest encryption algorithms for several advantages, for example:

- The ECC mainly based on discrete logarithm problem that known as Elliptic Curve Discrete Logarithm Problem (ECDLP) to determine the secret random number k from kp and p itself, and this is the secret of its strength unlike other algorithms such as RSA (Udin, et al., 2012).
- To compared the ECC with other public key algorithms, we see that, the ECC uses addition operations instead of multiplication and uses multiplication operations instead of exponentiation. (Tawalbeh, Mowafi and Aljoby, 2012).
- The ECC needs one-sixth the computational effort to get the same level of security that is required by RSA or Diffie-Hellman but with much shorter keys as shown in Table 3.4.
- The computational complexity strength needed for breaking Elliptic Curve

Encryption with a key of length 150 bits only. It is equivalent $3.8*10^{10}$ MIPS-years (i.e. MIPS-years millions of instructions per second times the required number of years) and $1.6*10^{28}$ MIPS-years if the ECC key length is increased to 234 bits . By using pollard's method (Stallings, 2011).

- It is used increasingly for wireless communications, smartcards because of the much smaller key sizes involved.

Table 3.4: Comparison of Achieve Equivalent Level of Security Depend on Different of Keys Sizes (Avinash Kak, 2012).

Symmetric Encryption Key Size in bits	RSA and Diffie-Hellman Key Size in bits	ECC Key Size in bits
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

To encrypt and decrypt the message during the ECC, we can only encrypt and decrypt set of points on the curve; also, we not divide the message into many blocks or stream. If sender A wants to send message M (plain text) securely to recipient B, Must do this method to represent message to a point (Encoding) and vice versa point to a message (Decoding) as shown in Figure 3.2.

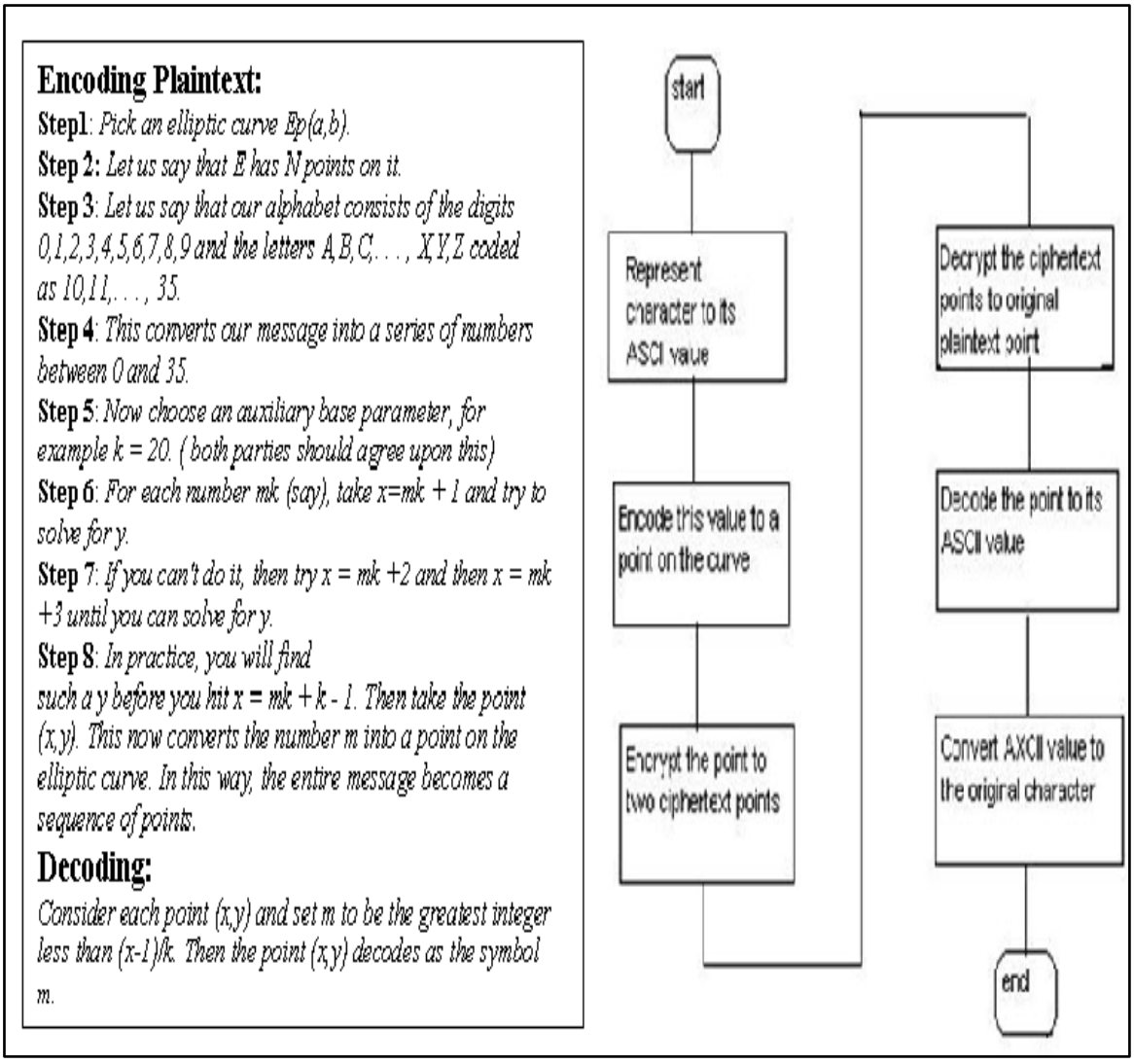

Figure 3.5: Koblitz's Method for Encoding and Decoding a Message (Padma Bh et. al.,2010)

2-Less-Complexity Encryption Algorithms

The medium-importance data will be encrypted using less-complexity encryption algorithms. Data encryption will be performed by powerful and less complex encryption algorithms such as, RSA Algorithm, Advanced Encryption Standard algorithm (AES), and Triple Data Encryption Standard algorithm (3DES). The data owner can use one or more algorithm according to his needs. Also each file is encrypted by a key or more different from the rest of the files in the same level.

As an example we can choose the RSA algorithm , which will be described below:

The RSA Algorithm

The RSA is type of Asymmetric Cryptography (or, Public Key Cipher) developed as a formal standard for encryption in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman (Rivest, Shamir and Adleman, 1977), and adoption as stand for their surname. RSA generates pair of protection keys called the public and private keys that are used to encrypt and decrypt the messages. RSA is one of the strongest encryption algorithms for several advantages (Wright,2007), for example:

- The RSA mainly based on the factoring problem (factoring large integers).
- RSA algorithm is the most widely used between public key algorithms.

Encryption in RSA

We let y = E(x) be the encryption function where x is an integer and y is the encrypted form of x

$$y = x^e \bmod n$$

Decryption in RSA

We let X = D(y) be the decryption function where y is an encrypted integer and X is the decrypted form of y

$$X = y^d \ mod\ n$$

3.2.3 Encrypted Data

This phase takes its input (the encrypted data) from the previous phase and will sort and give a unique number for each file, which is used for distinguish files and its levels, which is helps in retrieve and decrypt the data using the appropriate algorithm.

3.3 Case Study for Testing the Proposed Model.

The segmentation algorithms can be tested using any sample of data and as a case study; we will take a dataset of bank, which contains 12 columns and 600 rows. After that we will divide the dataset into three groups depends on its importance. To test the proposed algorithm, we build a simple application that will used to segment the dataset into three groups as shown in Figure 3.6.

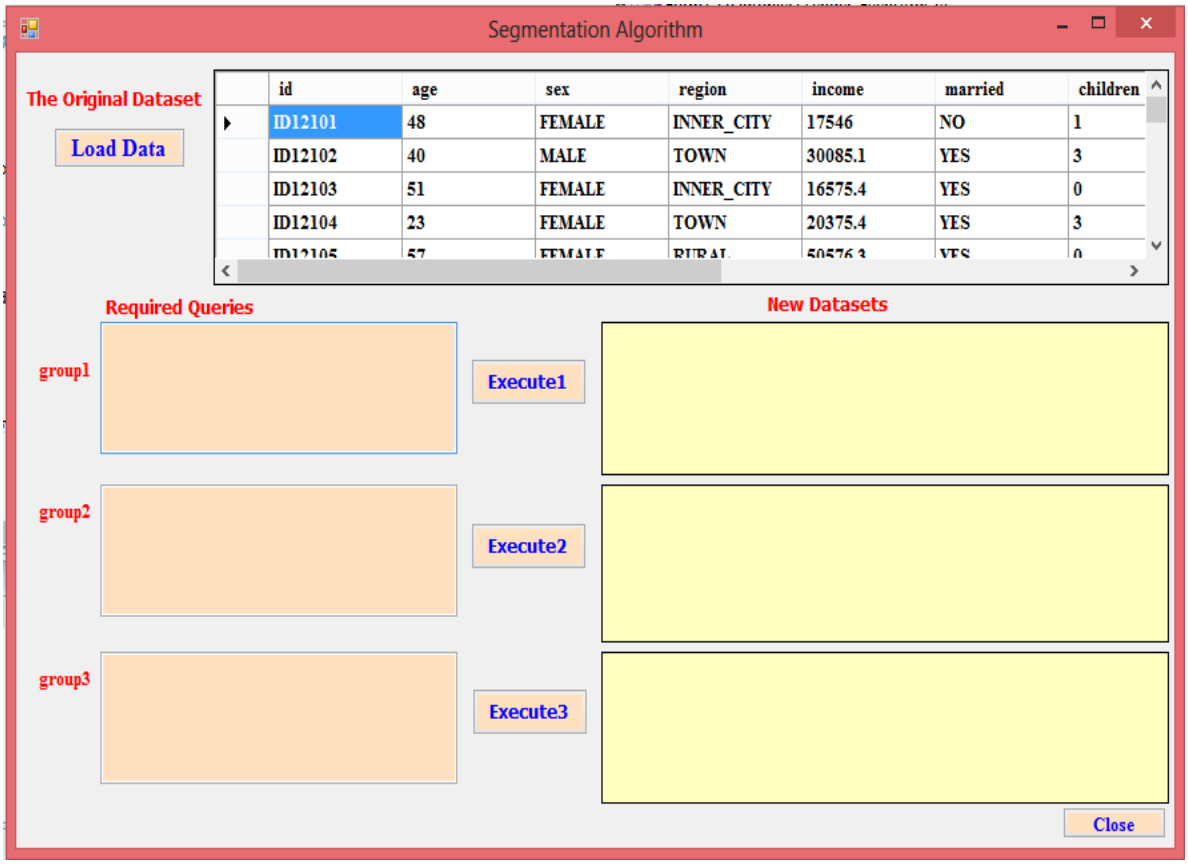

Figure 3.6: Retrieve the Required Data for Make Segmentation its.

As shown in the figure, the application consists from three main components which are:

- **Original Dataset:** display the required dataset (table or file).
- **Required Query:** consists of three parts, each of them contains a query to filter the data from the dataset for each level.
- **New Dataset:** the result of executing the queries on the original dataset

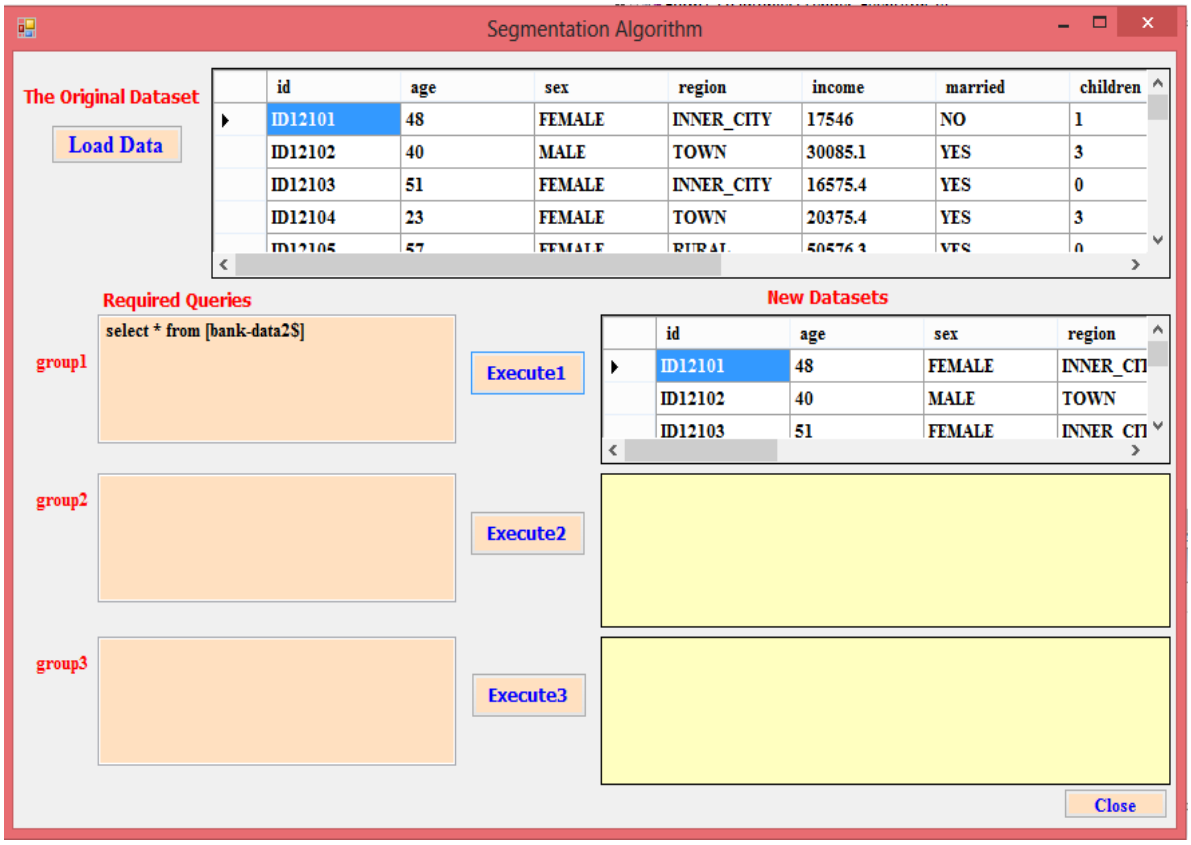

Figure 3.7: Execute Select Statement to Get on the Required Segmentation.

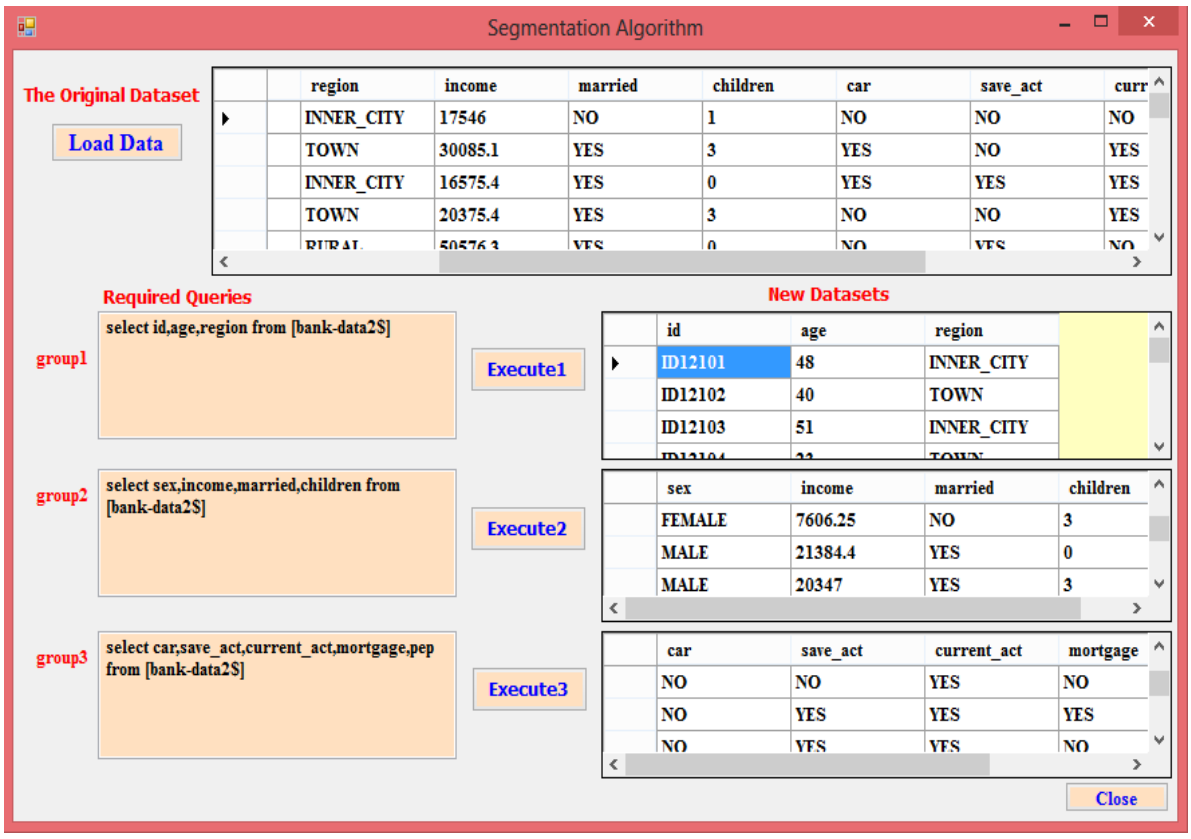

Figure 3.8: Execute Select Statement to Get on the Required Segmentation.

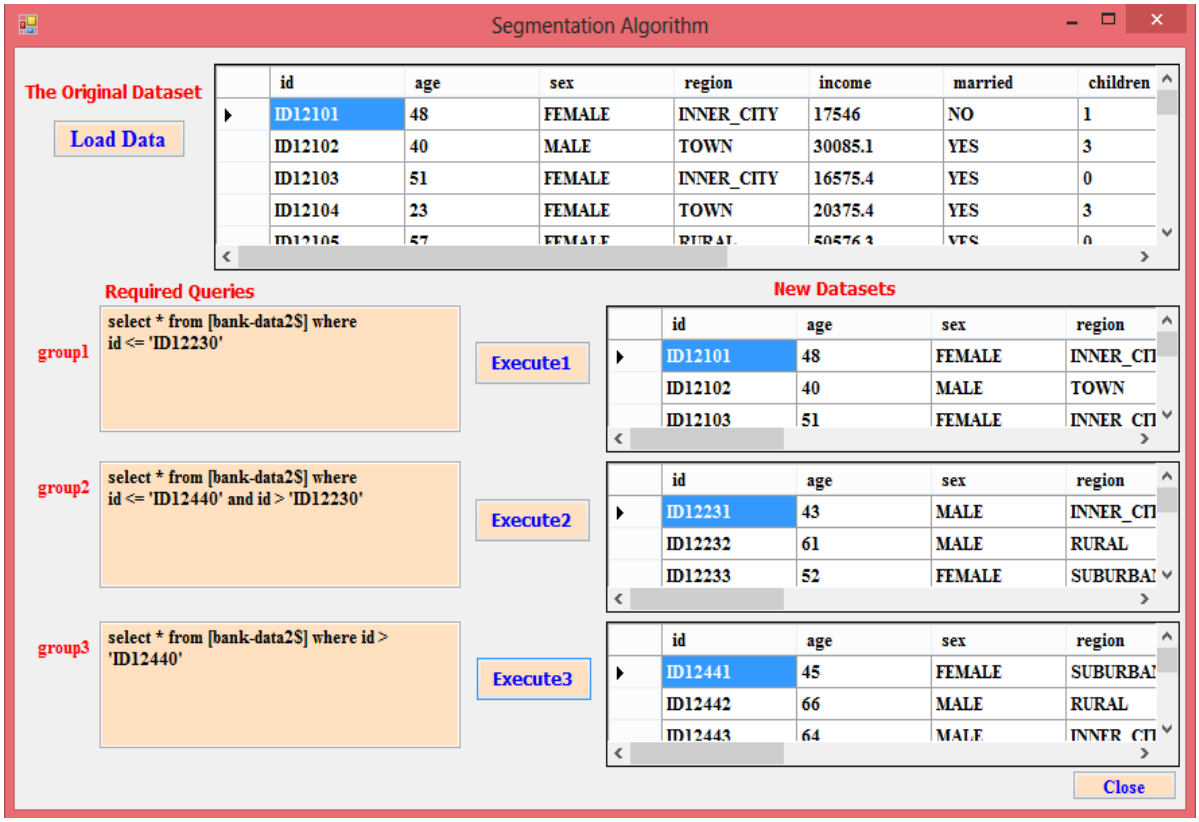

Figure 3.9: Execute Select Statement to Get on the Required Segmentation.

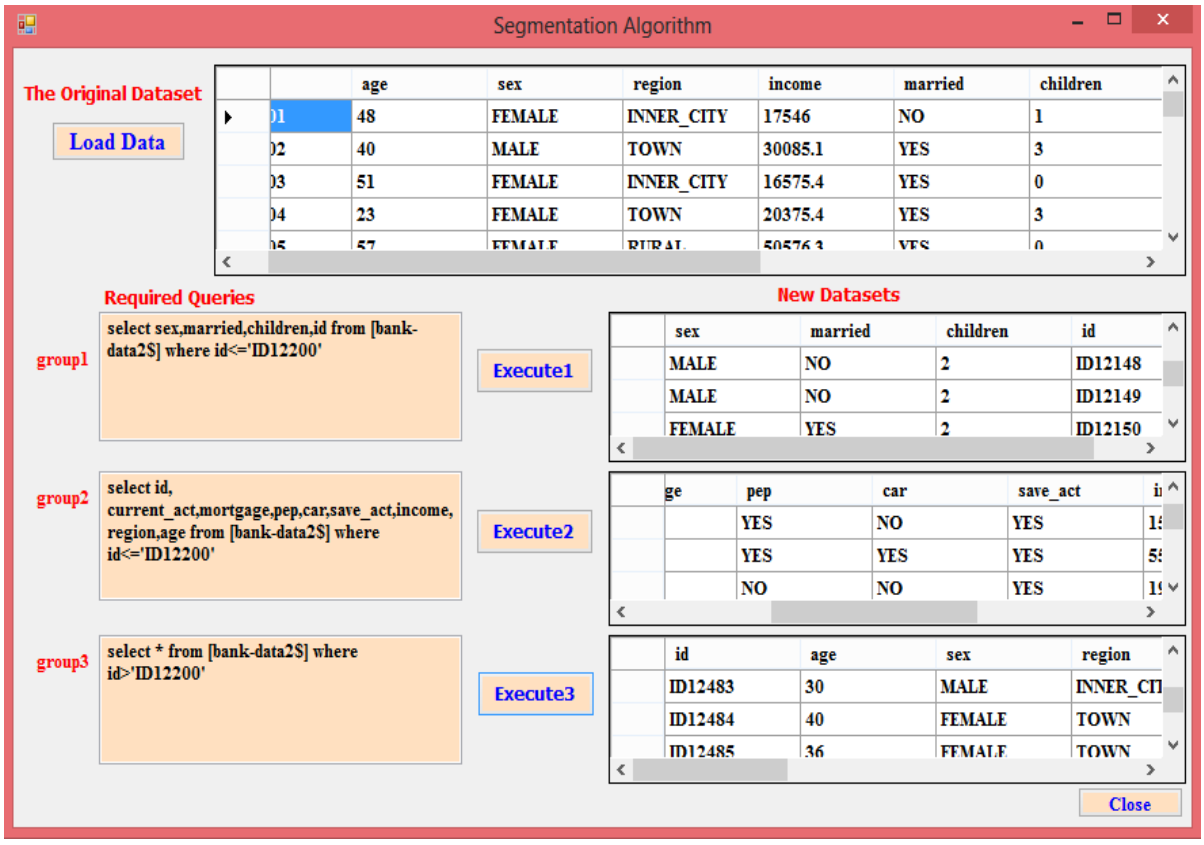

Figure 3.10: Execute Select Statement to Get on The Required Segmentation.

Chapter Four

Evaluation and Experimental Results

Chapter Four
Evaluation and Experimental Results

4.1 Introduction

This chapter presents detailed experiments about our model. The experiments were conducted on a dataset which will be covered in section 4.2. Section 4.3 shows the simulation environment to present and simplify understanding our idea. Section 4.4, gives details about the performance metrics that are used to evaluate our model under different conditions. Section 4.5 presents the experimental results that obtained from our model. Finally, section 4.6 presents the comparison to results that obtained from our model with other studies results.

4.2 Dataset Overview.

This section presents a brief idea about data that are used in our thesis where the data varies depending on need. We assumed that the dataset used in our work contain 147709 bytes before encryption. This dataset is separated into three levels (subsets) as shown in Table 4.1, by using Data-Segmentation Algorithm (DSA) which was explained in chapter three. The first level contains 50192 bytes which represents 30% of the original dataset. The second level is testing datasets that contains 50049 bytes and finally, third level contains 47468 bytes. The dataset size is not fixed and we can change it as required.

4.3 Simulation Environment and Procedures

According to our model architecture that explained in chapter three, it shows that each mentioned level in our architecture is logical according to its task**.** These levels need to be built on real software and subset of datasets needs to encrypt in each level. It is necessary to develop our model on a software platform that supports the

implementation of ECC, however we found some challenges. The first challenge was the infrastructure requirements such as hardware, software and the huge storage capacity. We need at least three computers to run three programs that execute encryption/decryption and they also should be connected with a private and public network. The second challenge was the software that should support the implementation of ECC. These challenges are the actual reason that directed us towards finding solutions by writing programming code to simulate our model**.**

Simulation is a simplified approach for presentation and evaluation of encryption algorithms under different levels. We execute our experiments using a laptop with a specifications of 2.99 GHz CPU and 2 GB RAM. The laptop is used to encrypt a sample of data that is explained in section 4.2. The "MATLAB R2010a" software was used to encrypt / decrypt data in each level depending on different keys**.** We are trying to find the comparison between data before and after segmentation in many aspects such as size, time and security.

4.4 Performance Evaluation Metrics

This section aims to present the measurements that are used to measure the accuracy of our model**.** These measurements are used to obtain the efficiency and effectiveness of the encryption algorithms that are used in each level and also to compare changing in data size before and after segmentation**.** Three measurements are used to evaluate the accuracy and performance of our model and are described below**:**

- **Calculate Dataset Size Before Encryption.**

It aims to calculate the original size (in Bytes) of the dataset that we aim to encrypt and store in the Cloud. Figure 4.1 shows a program written in MATLAB to calculate the size of any file.

```
function [PlainText_Point] = ECCtext

fid=fopen('bank10.txt','r'); %% read dataset before encryption
Array=fscanf(fid,'%c');
fclose(fid);
fid1 = fopen('results.txt', 'wt'); %% file to write the results after encryption
tic
for l=1:1:length(Array)

[point] = Message_Encode(Array(l));

[Ciphered_point1 , Ciphered_point2] = Message_Encryption(point(1) , point(2));

fprintf(fid1,'%d','%d','%d ','%d ',Ciphered_point1(1),Ciphered_point1(2),Ciphered_point2(1),Ciphered_point2(2));
fprintf(fid,'%s\n ',' ');

[PlainText_Point] = Message_Decryption([Ciphered_point1(1) Ciphered_point1(2)],[Ciphered_point2(1) Ciphered_point2(2)]);
Plaintext = Message_Decode(PlainText_Point(1));

Plaintext= char(Plaintext);

end
time=toc
time= time * 1000;

f=dir('bank10.txt');%% read the desription of file
s=f.bytes; %% choose only the size in byte
```

Figure4.1:**MATLAB code to calculate the dataset size (bank10.txt) before encryption**

- ## Calculate Dataset Size After Encryption.

This measure calculates the size (in Bytes) of dataset after encryption. Figure 4.2 shows a program written in MATLAB to calculate the size of any ciphered file.

```
function [PlainText_Point] = ECCtext

fid=fopen('bank10.txt','r'); %% read dataset before encryption
Array=fscanf(fid,'%c');
fclose(fid);
fid1 = fopen('results.txt', 'wt'); %% file to write the results after encryption
tic
for l=1:1:length(Array)

[point] = Message_Encode(Array(l));

[Ciphered_point1 , Ciphered_point2] = Message_Encryption(point(1) , point(2));

fprintf(fid1,'%d','%d','%d ','%d ',Ciphered_point1(1),Ciphered_point1(2),Ciphered_point2(1),Ciphered_point2(2));
fprintf(fid,'%s\n ',' ');

[PlainText_Point] = Message_Decryption([Ciphered_point1(1) Ciphered_point1(2)],[Ciphered_point2(1) Ciphered_point2(2)]);
Plaintext = Message_Decode(PlainText_Point(1));

Plaintext= char(Plaintext);

end
time=toc
time= time * 1000;

f=dir('results.txt');
s=f.bytes;
fprintf('Size of dataset in bytes after encryption is %f\n',s); %% size of ciphered file
fclose(fid1);
```

Figure 4.2: MATLAB Code to calculate the dataset size (bank10.txt) after encryption

- **Calculate The Encryption / Decryption Time**

The following program calculates the time (in millisecond) elapsed in the encryption / decryption process. Figure 4.3 shows a program written in MATLAB to calculate the required time for this measure.

```
end
time=toc % read the Elapsed Time
time= time * 1000; % read the Elapsed Time in millisecond

f=dir('bank10.txt');%% read the desription of file
s=f.bytes; %% choose only the size in byte

fprintf('the dataset name is: Bank1 %.f\n');
fprintf('Elapsed Time in millisecond is %.f\n',time);
fprintf('Size of dataset in bytes before encryption is %f\n',s); %% size of raw file

f=dir('results.txt');
s=f.bytes;
fprintf('Size of dataset in bytes after encryption is %f\n',s); %% size of ciphered file
fclose(fid1);

 tic
 [Pk]= GET_Public_Key('A');
 Time_kG=toc;
 Time_kG= Time_kG * 1000;
 fprintf(' Time to generate pk in millisecond is %.f\n',Time_kG);
```

Figure 4.3: MATLAB Code to calculate the elapsed time for encryption/decryption

4.5 Experimental Results

This section explains the experimental results to verifying the performance of our model for encryption and storing data in the Cloud. It also presents and discusses the results of our simulation. We will compare data before and after segmentation, according to metrics mentioned in the previous section (4.4). The results will be presented by using figures, charts and tables.

4.5.1 The non-Segmented vs. Segmented Data

We started our experiment by choosing samples of datasets (randomly) which contain 147709 bytes (as an example). Figure 4.4 shows the size of the dataset before and after segmentation.

```
>> ECCtext

time =

  150.4062

the dataset name is: allBank
Elapsed Time in millisecond is 150406
Size of dataset in bytes before segmentation is 147709.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 147709
Size of dataset in bytes after encryption is 8081816.000000
 Time to generate pk in millisecond is 0

ans =

    11600006        6144
```

Figure 4.4: Size of the data before and after segmentation

We use the Data-Segmentation Algorithm (DSA) that mentioned in chapter three to get three groups (subsets) of the original dataset as shown in table 4.1.

Table 4.1. Size of dataset before and after segmentation

Dataset name	Size after segmentation in byte	Size before segmentation in byte
Bank1	**50192**	**147709**
Bank2	**50049**	
Bank3	**47468**	
	Total : 147709	**Total : 147709**

From Table 4.1 we can note that total data size before and after segmentation is equal as shown in Figure 4.5 but this size will change after encryption.

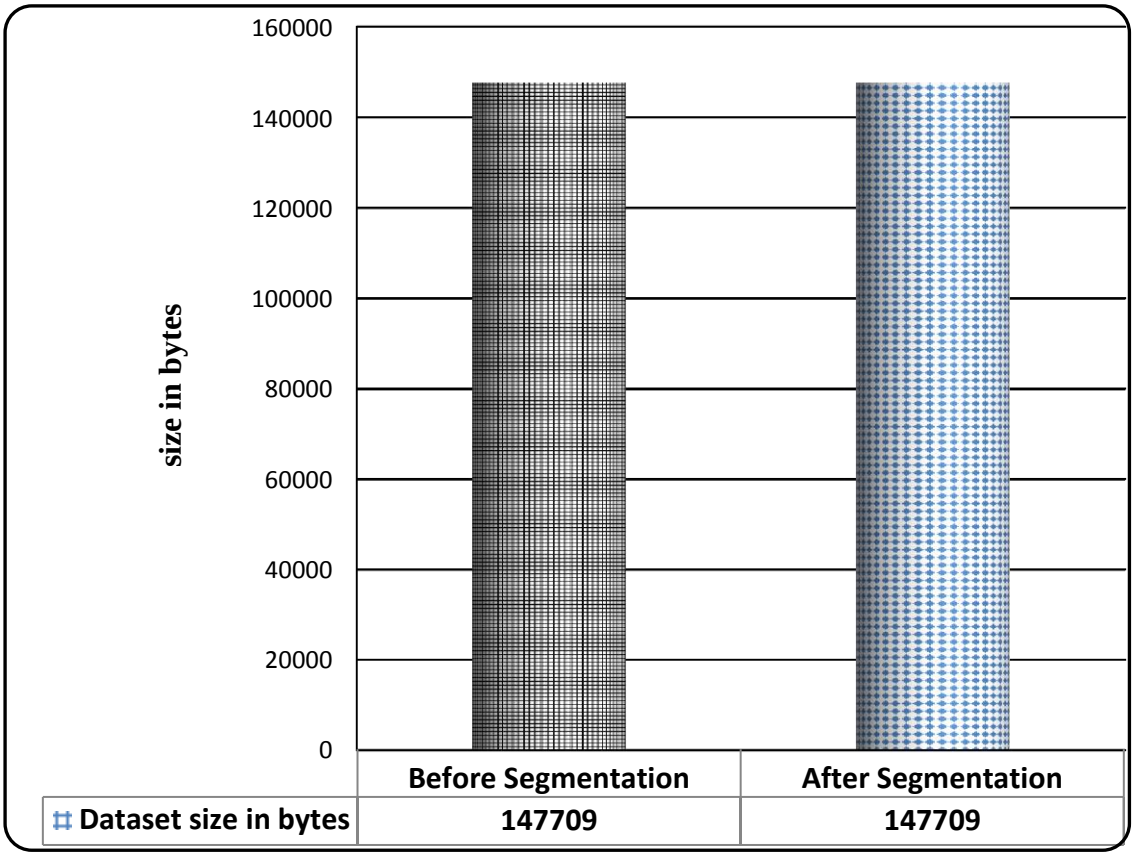

Figure 4.5: Dataset size before and after segmentation

The second step is using "MATLAB" to execute the ECC algorithm with varying parameters values in order to get different encryption keys which can be used in different levels. According to the idea in our model in (chapter three) the data is encrypted in level one and level two while there is no need for encryption the data in level three, each level has a different size. The following Figures 4.6 and 4.7 show the results when the 100 bytes encryption key is used.

```
>> ECCtext

time =

   52.1419

the dataset name is: Bank1
Elapsed Time in millisecond is 52142
Size of dataset in bytes before segmentation is 50192.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 50192
Size of dataset in bytes after encryption is 2746221.000000
 Time to generate pk in millisecond is 0

ans =

     4400012        3456

>>
```

Figure 4.6: Results of using 100 bytes key to encrypt dataset in First Level

```
>> ECCtext

time =

   53.1423

the dataset name is: Bank2
Elapsed Time in millisecond is 53142
Size of dataset in bytes before segmentation is 50049.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 50049
Size of dataset in bytes after encryption is 2738386.000000
 Time to generate pk in millisecond is 0

ans =

     4600037        5120
```

Figure 4.7: Results of using 100 bytes key to encrypt dataset in Second Level

Figures 4.6 and 4.7 summarize the results of using 100 bytes key to encrypt dataset. Table 4.2 shows the difference in the size of data before and after encryption**.** Figure 4.8 shows the results of data size in each level after segmentation and encryption.

Table 4.2 : Dataset size before and after segmentation with pr_k_A = 100

Dataset name	Data Size Before Encryption in Bytes	Data Size After Encryption in Bytes
Bank1	50192	2746221
Bank2	50049	2738386
Bank3	47468	47468
	Total size before EN	Total size after EN
	147709	5532075

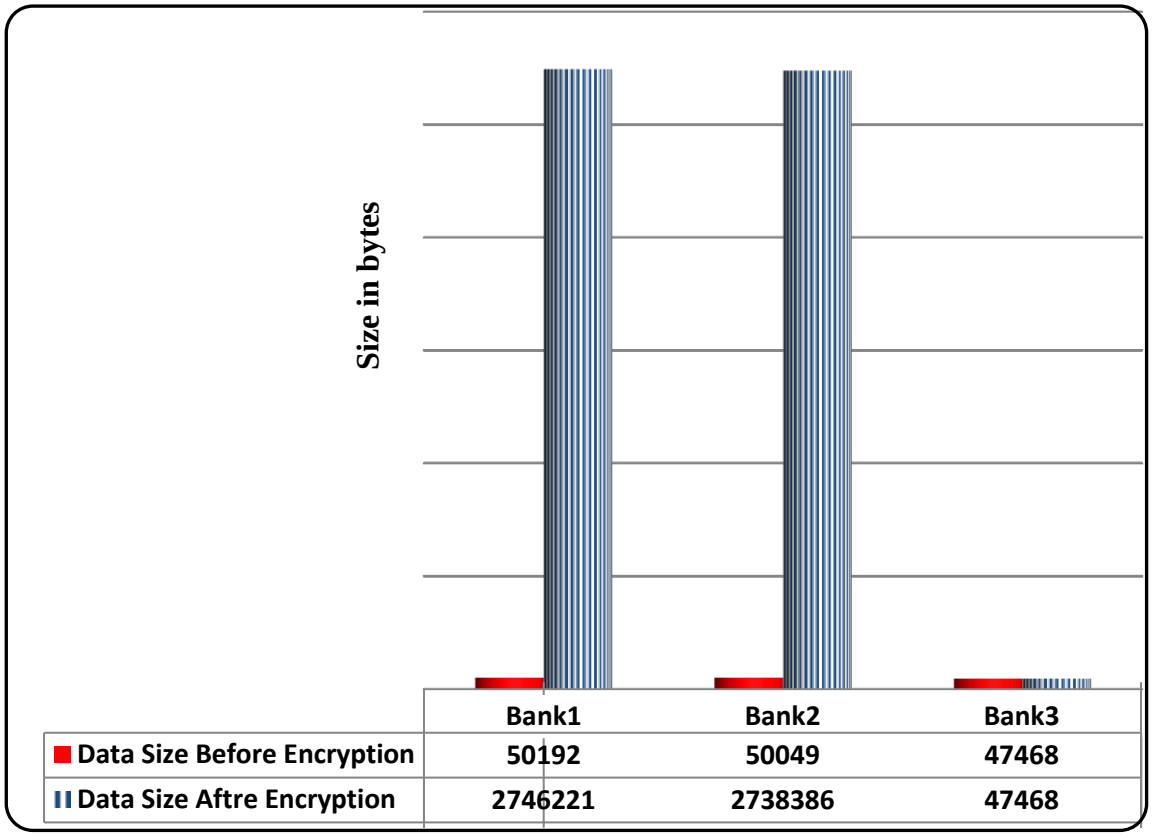

Figure 4.8: The comparison of the size after segmentation with pr_k_A=100

Depending on the experimental results which executed with pr_k_A = 100 , we can note that the data size before segmentation and after encryption is bigger than its size after segmentation and encryption as shown in Table 4.3 and Figure 4.9.

Table 4.3. Experimental results for the chosen dataset with pr_k_A = 100.

	Size Before Encryption	Size After Encryption
Before Segmentation	**147709**	**8081816**
After Segmentation	**147709**	**5532075**

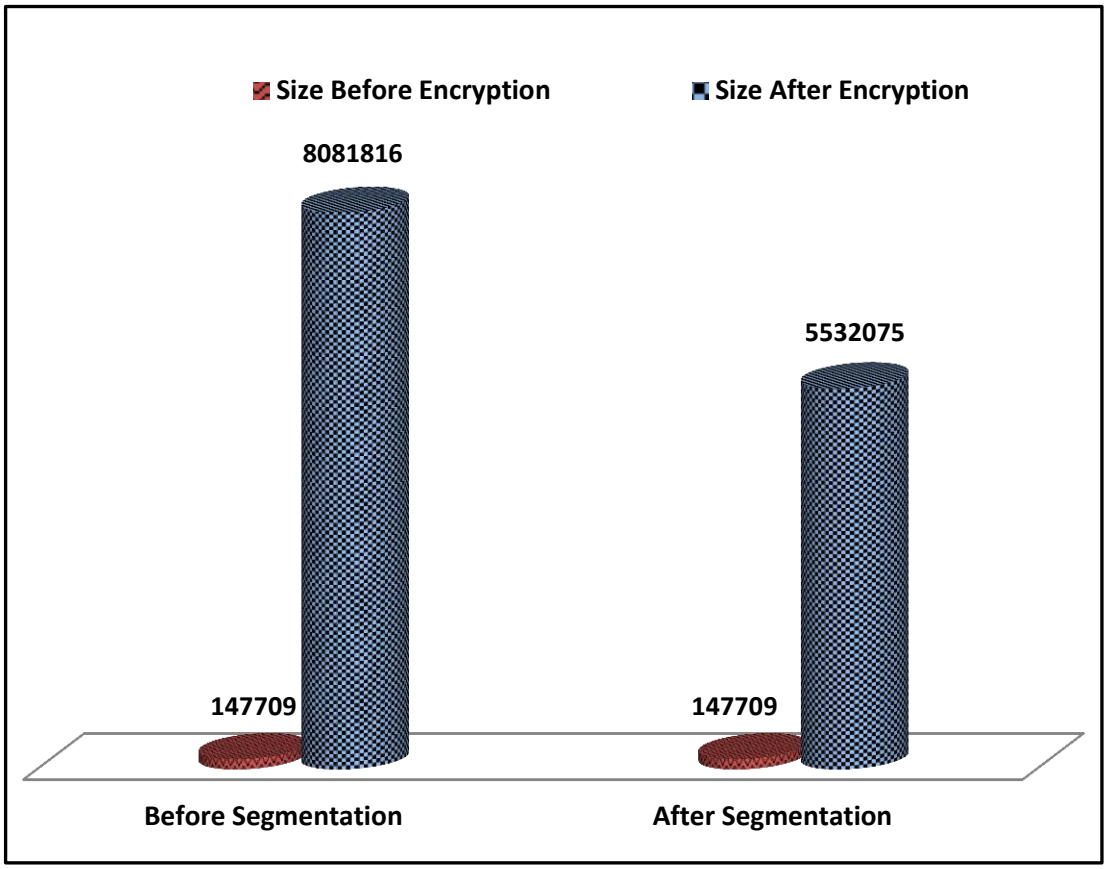

Figure 4.9: Dataset size before and after segmentation with pr_k_A = 100.

From the previous Figure 4.9, we can note that our scheme presents better results after segmentation and encryption.

The next experimental results show the difference in size of the dataset in case of changing the size of encryption key. Figure 4.10 shows the results of using 50 bytes as an encryption key.

```
>> ECCtext

time =

  124.6093

the dataset name is: allBank
Elapsed Time in millisecond is 124609
Size of dataset in bytes before segmentation is 147709.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 147709
Size of dataset in bytes after encryption is 8049330.000000
 Time to generate pk in millisecond is 0

ans =

    45810606    27442476
```

Figure 4.10 : The dataset size with pr_k_A = 50

We will illustrate the results of data size in each level after segmentation in case using the encryption key of size = 50 bytes, as shown in figure 4.11 (for level one) and in figure 4.12 (for level two)**.**

```
>> ECCtext

time =

   47.6034

the dataset name is: Bank1
Elapsed Time in millisecond is 47603
Size of dataset in bytes before segmentation is 50192.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 50192
Size of dataset in bytes after encryption is 2735287.000000
 Time to generate pk in millisecond is 0

ans =

    45002870     6033804
```

Figure 4.11: Dataset size in level-1 when pr_k_A=50

```
>> ECCtext

time =

   42.7688

the dataset name is: Bank2
Elapsed Time in millisecond is 42769
Size of dataset in bytes before segmentation is 50049.000000
the dataset name is: Bank1,bank2 and bank3
Size of dataset in bytes after segmentation is 50049
Size of dataset in bytes after encryption is 2727791.000000
 Time to generate pk in millisecond is 0

ans =

    25374138    30031034
```

Figure 4.12: Dataset size in level-2 when pr_k_A =50

From figure 4.11 and figure 4.12 we can see the data size results in each level after segmentation when we use the encryption key of size = 50 bytes, these results are summarized as shown in table 4.4 and figure 4.13.

Table 4.4 : Dataset size After Segmentation with pr_k_A = 50

Dataset name	Data Size Before Encryption in Bytes	Data Size After Encryption in Bytes
Bank1	**50192**	**2735287**
Bank2	**50049**	**2727791**
Bank3	**47468**	**47468**
	Total size before EN	**Total size after EN**
	147709	**5510546**

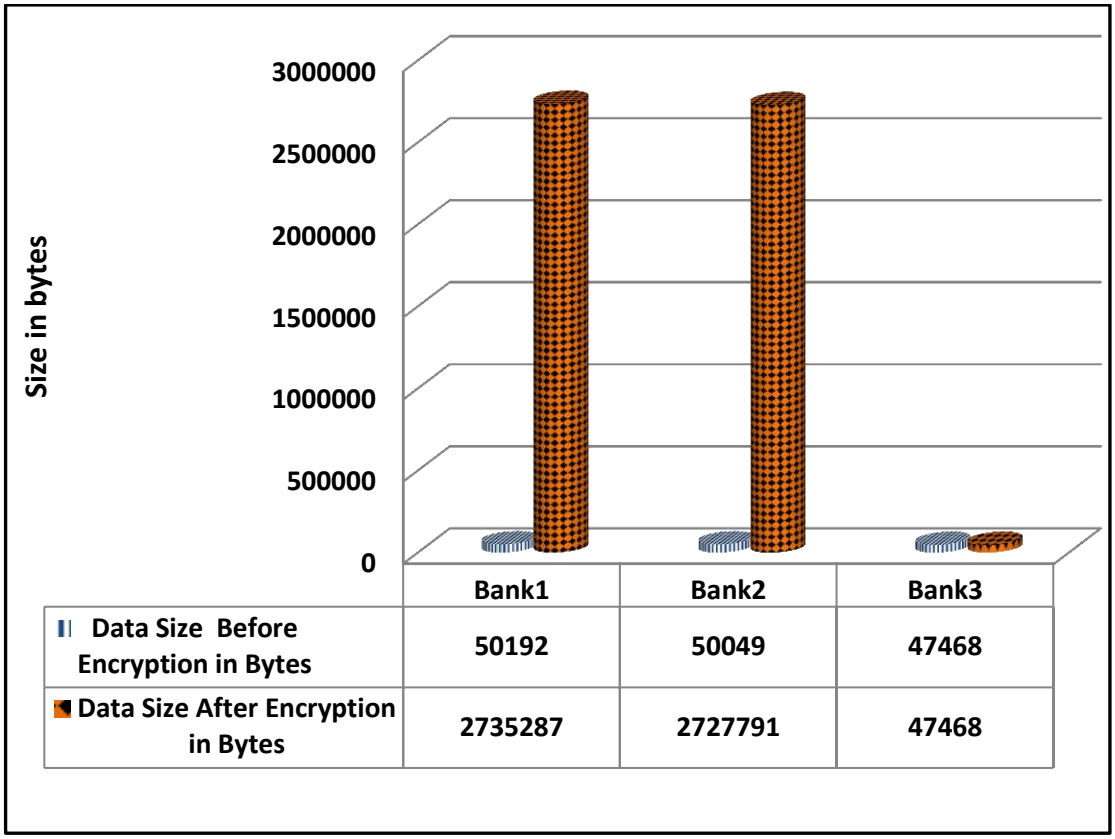

Figure 4.13 : Dataset size in levels with pr_k_A =50

Also, we compared between the results of both encryption keys that are used before, as shown in table 4.5 and figure 4.14.

Table 4.5: The comparison of the dataset size with pr_k_A=100 and pr_k_A=50.

	Before segmentation	After segmentation
Size of dataset with Pr_k_A = 100	8081816	5532075
Size of dataset with Pr_k_A = 50	8049330	5510546

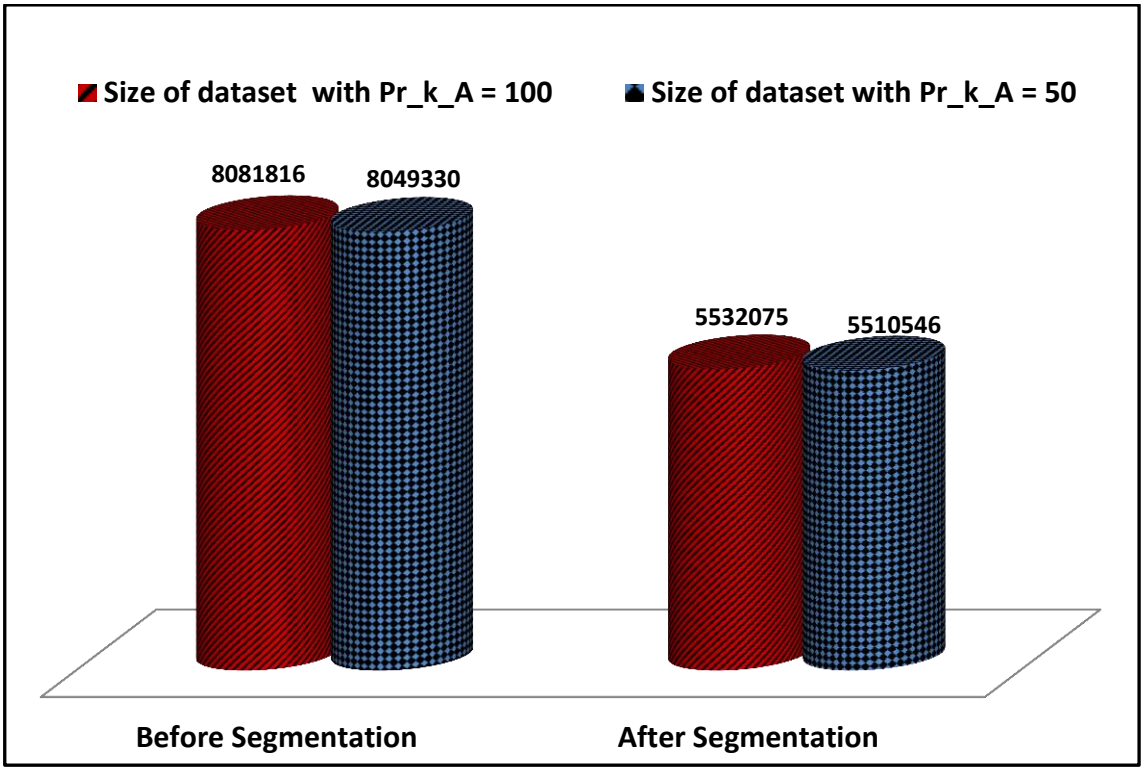

Figure 4.14: Comparison between the dataset size with pr_k_A=100 and pr_k_A=50.

Table 4.6 and Figure 4.15, shows a comparison between the dataset size when each level uses different encryption key.

Table 4.6: The dataset size in each level with different keys (pr_k_A=100 and 50).

Dataset name	Data Size Before Encryption in Bytes	Data Size After Encryption in Bytes when pr_k_A =100	Data Size After Encryption in Bytes when pr_k_A =50
Bank1	**50192**	**2746221**	2735287
Bank2	**50049**	2738386	**2727791**
Bank3	47468	47468	47468
	Total size before EN	Total size after EN	Total size after EN
	147709	5532075	5510546

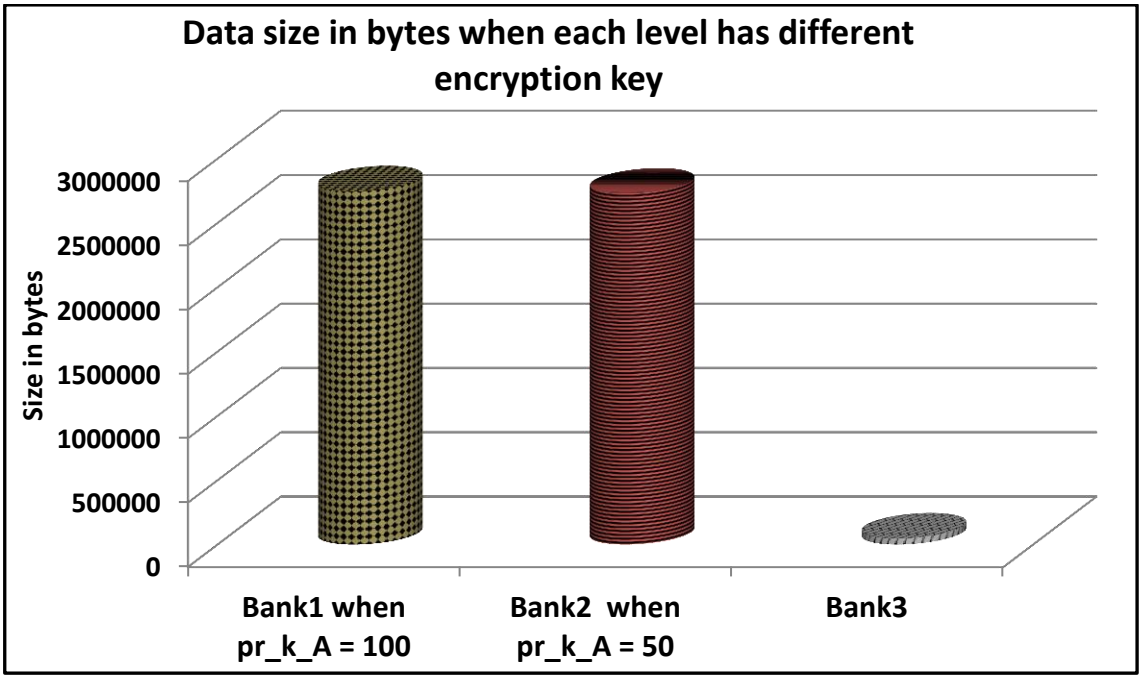

Figure 4.15: Comparison the dataset size for levels with different keys (pr_k_A=100 and 50).

To measure elapsed time in the encryption process, we will explain it in the following experiments. The time required for encryption before and after segmentation when we use pr_k_A = 100 bytes as shown in table 4.7 and table 4.8 respectively.

Table 4.7 : The time of encryption before segmentation with pr_k_A = 100

Dataset name	Size before encryption in byte	Size after encryption in byte	Encryption time in ms
Bank	**147709**	**8081816**	**150406**

Table 4.8: The time of encryption after segmentation with pr_k_A = 100.

Dataset name	Size before Encryption in byte	Size after Encryption in byte	Encryption time in ms
Bank1	**50192**	**2746221**	**52142**
Bank2	**50049**	**2738386**	**53142**
Bank3	**47468**	**47468**	**0.0**
	Total size before EN	**Total size after EN**	**Total-time in ms**
	147709	**5532075**	**105284**

Figure 4.16 shows the results of the time spent for encryption in each level with key = 100 bytes, according to our model.

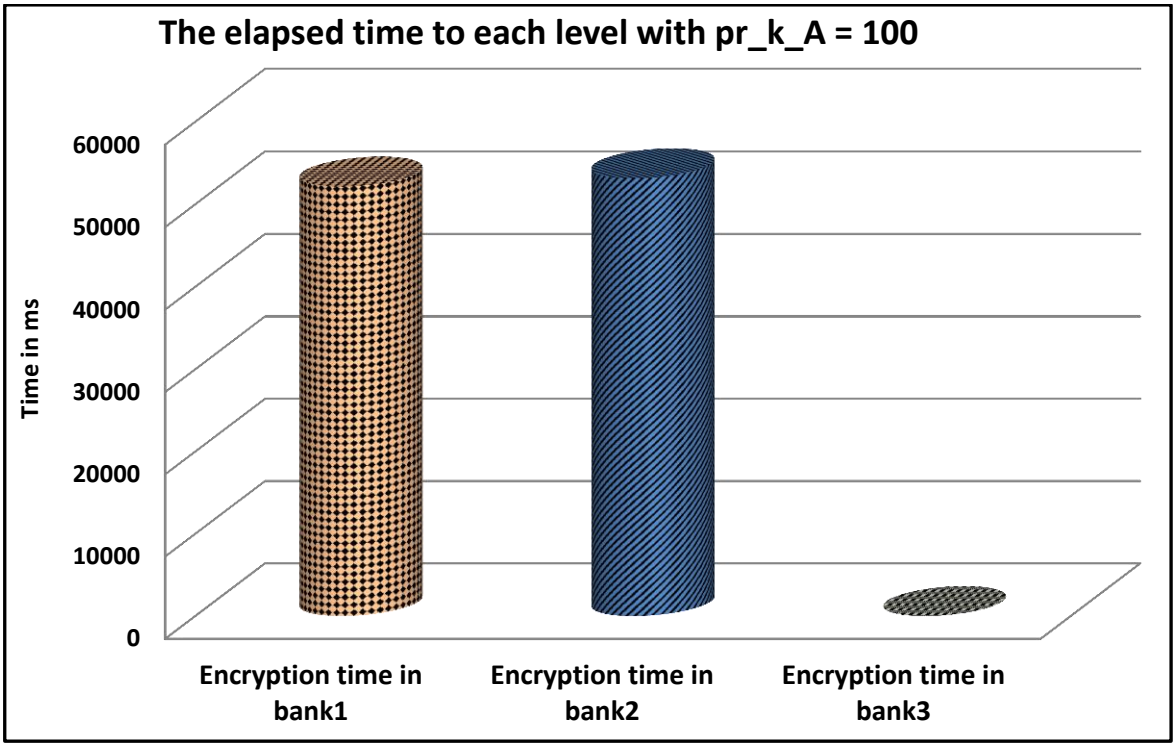

Figure 4.16: The elapsed time to each level with pr_k_A = 100.

We can also compare between the results of the elapsed time to encrypt data before and after the segmentation as shown in figure 4.17. The results shows that the performance of our model better than performance of encryption without segmentation.

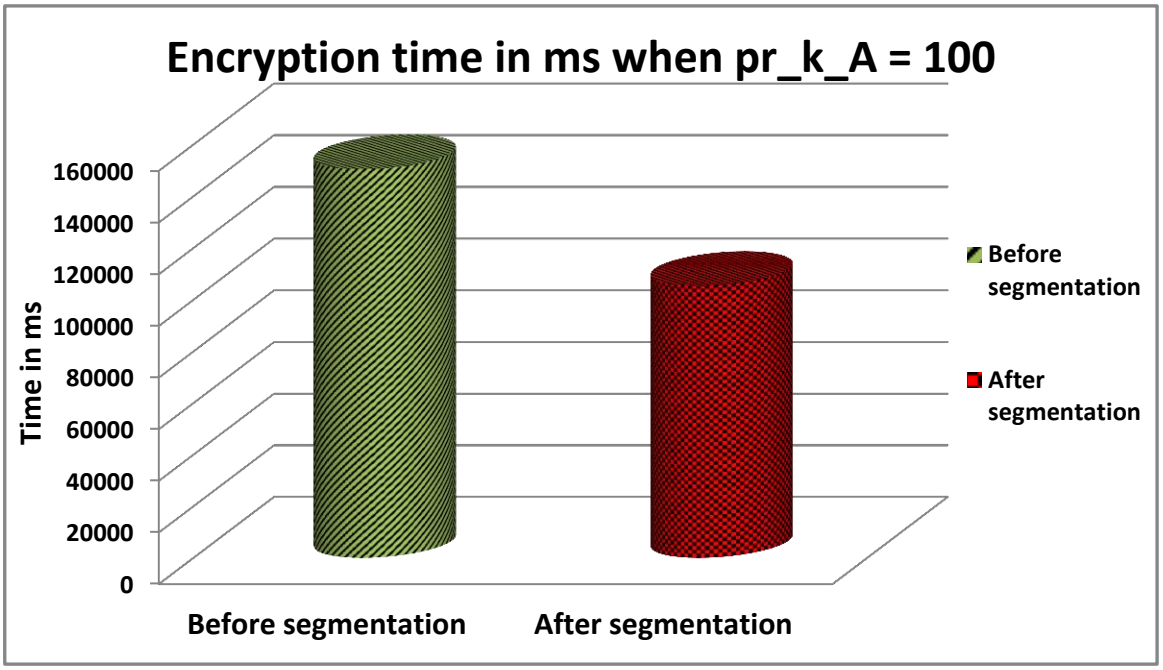

Figure 4.17: The comparison of the Encryption time with pr_k_A = 100.

Also table 4.9 and table 4.10 shows the time required for encryption before and after segmentation when we use pr_k_A = 50.

Table 4.9: The time of encryption before segmentation with pr_k_A = 50

Dataset name	Size before encryption in byte	Size after encryption in byte	Encryption time in ms
Bank	147709	8049330	124609

Table 4.10: The time of encryption after segmentation with pr_k_A = 50.

Dataset name	Size before encryption in byte	Size after encryption in byte	Encryption time in ms
Bank1	50192	2735287	47603
Bank2	50049	2727791	42769
Bank3	47468	47468	0.0
	Total size before EN	Total size after EN	Total-time in ms
	147709	5510546	90372

Figure 4.18 presents the results of the time spent for encryption in each level according to our model previously when we use pr_k_A = 50.

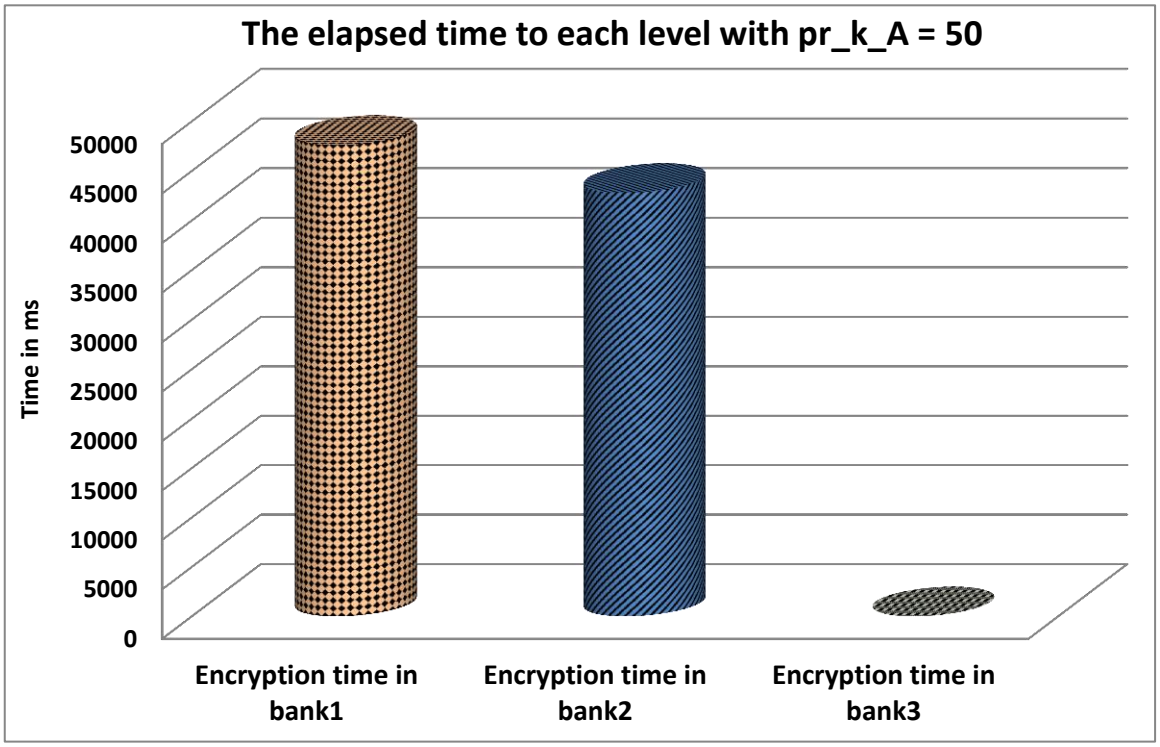

Figure 4.18: The elapsed time to each level with pr_k_A = 50.

The time required for encryption before and after segmentation when we use

pr_k_A = 50 bytes as shown in table 4.11 and figure 4.19 respectively.

Table 4.11: Comparison between the required encryption time with pr_k_A= 50.

	Before segmentation	After segmentation
Encryption time in ms	**124609**	**90372**

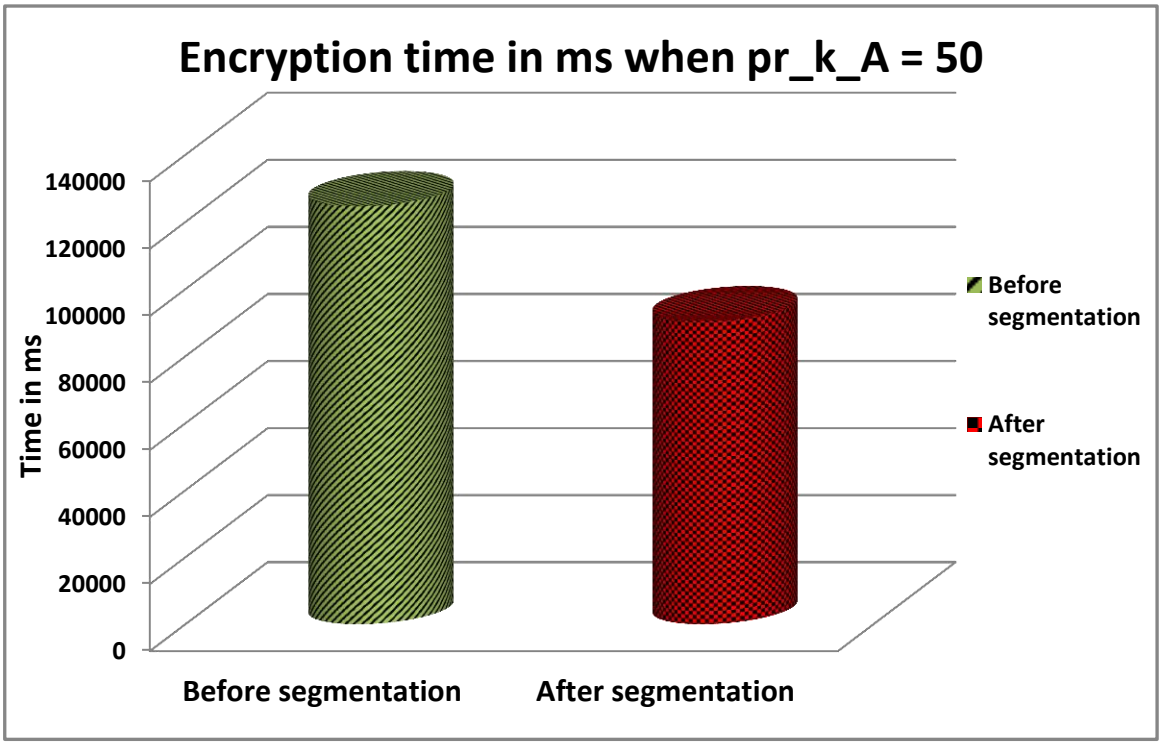

Figure 4.19: Comparison between the required encryption time with pr_k_A = 50.

From previous experiments which are used to measure the elapsed time at encryption we can note that the time is increased when the size of data is increased. Also, we can note that the time is decreased when the same data size is fragmented as shown in table 4.12 and in figure 4.20.

Table 4.12: Comparison of the required encryption time with pr_k_A = 100 and pr_k_A = 50.

	Encryption time in ms with pr_k_A = 100	Encryption time in ms with pr_k_A = 50
Before segmentation	**150604**	**124609**
After segmentation	**105284**	**90372**

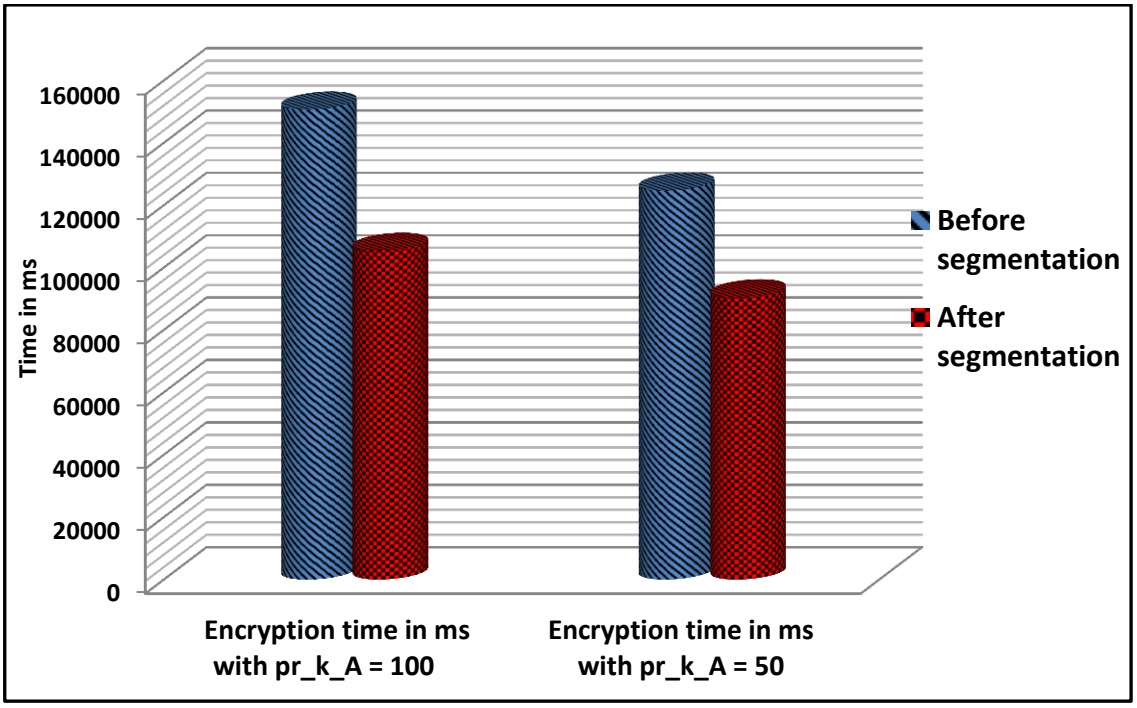

Figure 4.20: Comparison of the required encryption time with pr_k_A = 100 and pr_k_A = 50.

Table 4.13 and Figure 4.21 show a comparison between the required encryption time when each level uses encryption key with different size, as example the first level uses encryption key equal 100 bytes while the second level uses encryption key equal 50 bytes.

Table 4.13: Comparison of the required encryption time when each level has different encryption key.

Dataset name	Encryption time in ms when pr_k_A =100	Encryption time in ms when pr_k_A =50
Bank1	**52142**	47603
Bank2	53142	**42769**
Bank3	0.0	**0.0**
	Total time before EN	Total time after EN
	105284	90372

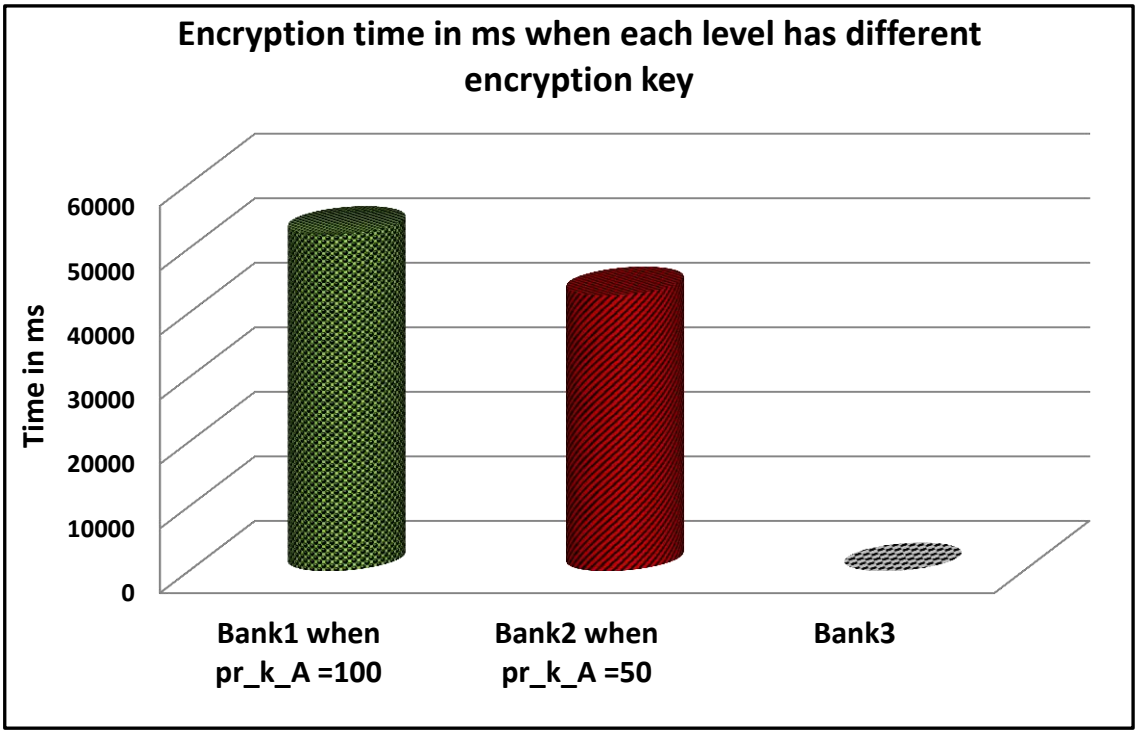

Figure 4.21: The required encryption time when each level has different encryption key.

Chapter Five
Conclusions and Recommendations for Future Research

Chapter Five
Conclusions and Recommendations for Future Research

5.1 Introduction.

This chapter summarizes the conclusions of our work model and the suggested recommendations for future work. In section 5.2 we present the main conclusion from our model. While, section 5.3. is discussing some future directions and suggesting to gain more improvements in an encryption domain on the Cloud Computing.

5.2 Conclusions.

According to the goals and the experimental results we can summarize our contributions as follows:

- In our work model we present modifications on the current encryption models in Cloud Computing to increase the data security.
- Our model builds a multi-level architecture which reflects the required encryption algorithms in each level.
- The Data-Segmentation Algorithm (DSA) is used in our model in order to partitioning the large data size into proper and manageable sets depending on specific criteria.
- Using the MATLAB to programming and a simulating, as well as, using the Elliptic Curve Cryptography (ECC) to implementation. In order to shows that our model is applicable.
- Our model achieved best results than other models that is used in the encryption domain within Cloud Computing.

5.3 Recommendations for Future Research.

While working in this thesis; there are some ideas and questions where generated in order to continues of developing and enhancing our model. We can suggest some ideas for future research which focus on the following**:**

- Upgrading our model architecture in order to remove any hindrances in terms of efficiency and effectiveness**.**
- Using encryption algorithms other than the used in our model in order to see different the implementation in more optimal ways which addresses issues of time and reduction cost**.**
- Implement the model with applications which supports large key's size different with the Elliptic Curve Cryptography (ECC) which is supports limited key's size**.**

References

Atayero A. & Feyisetan O. (2011), 'Security Issues in Cloud Computing: The Potentials of Homomorphic Encryption', *Journal of Emerging Trends in Computing and Information Sciences*, vol. 02, no. 10, pp. 546-552, October 01.

Chen D. & He Y. (2010), 'A Study on Secure Data Storage Strategy in Cloud Computing', *Journal of Convergence Information Technology* , vol. 05, no. 07, pp. 43-49 , September 01.

Cloud Security Alliance. (2010), 'Top Threats to Cloud Computing V1.0', *Cloud Security Alliance*, March, <https://Cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>.

Enterprise systems,< http://esj.com/articles/2012/02/16/porticors-virtual-private-data-system.aspx /> , viewed by 16 February 2012.

Kumbhar N., Chaudhari V. & Badhe M. (2012), 'The Comprehensive Approach for Data Security in Cloud Computing: A Survey', *International Journal of Computer Applications (0975 – 8887)*, Foundation of Computer Science, New York, USA, vol. 39, no. 18, pp. 23-29, February.

Mell P. & Grance T. (2011), 'The NIST Definition of Cloud Computing', *National Institute of Standards and Technology*, Gaithersburg, USA, September.

Nexsan 2011, < http://www.nexsan.com/products/assureon/library/Assureon_DS.pdf />, viewed by 13 Sept 2011.

Pagano F. & Pagano D. (2011), 'Using In-Memory Encrypted Databases on the Cloud', *Securing Services on the Cloud (IWSSC), 2011 1st International Workshop on* , IEEE, pp. 30 - 37, September 6-8.

Plummer D., Bittman T., Austin T., Cearley D., & Smith D. (2008), 'Cloud Computing Defining and Describing an Emerging Phenomenon', *Gartner*, June 17.

Puttaswamy K .P., Kruegel C. & Zhao B. (2010), 'Protecting Private Data in Third-Party Compute Cloud s', *Poster at the Proceedings of HotMobile 2010*, ACM,vol. 02, no. 06, pp. 2150-2159.

Bh P., Chandravathi D. & Roja P. P. (2010), ' Encoding And Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz's Method', *International Journal on Computer Science and Engineering (IJCSE)*, Chennai, India, vol. 02 , no. 05 , pp. 1904-1907, September.

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

Foundation of Computer Science, New York, USA, vol. 34, no. 09, pp. 43-49 , November.

Shen E., Shi E. & Waters B. (2009), 'Predicate Privacy in Encryption Systems', *TCC '09 Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography,* Springer-Verlag , Berlin, Heidelberg , pp. 457 - 473 .

VMWare2009. *Virtualization basics*. <http://www.vmware.com/technology/virtual-machine.html >.

Yuefa D., Bo W., Yaqiang G., Quan Z. & Chaojing T. (2009), 'Data Security Model for Cloud Computing', *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009),* Qingdao, China, November 21-22.

Buyya R., Yeo C., Venugopal S., Broberg J., & Brandic Iv (2009), 'Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility ', *Future Generation Computer Systems*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, vol. 25 , no. 6 , pp. 599-616, June .

Qaisar S. & Khawaja K. (2012), 'Cloud Computing: Security Threats and CounterMeasures', *Interdisciplinary Journal of Contemporary Research in Business (IJCRB),* Institute of Interdisciplinary Business Research , vol. 3 , no. 9 , USA , January .

Madhavi K., Tamilkodi R., & Sudha K. (2012), 'Cloud Computing: Security Threats and Counter Measures', *International Journal of Research in Computer and Communication technology (IJRCCT),* vol. 1 , no. 4 , USA , September .

Singh A. & Shrivastava M. (2012), 'Overview of Attacks on Cloud Computing ', *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 1 , no. 4 , USA , April .

Singh A. & Shrivastava M. (2012), ' Overview of Security issues in Cloud Computing ', *International Journal of Advanced Computer Research (IJACR),* Association of Computer Communication Education for National Triumph , vol. 2 , no. 1 , pp. 41-45 , March .

MIT, (1999). *Lincoln Laboratory,* (On-Line), Available: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html

Bhadauria R., & Sanyal S. (2012), 'Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques ', *International Journal of Computer Applications*, Foundation of Computer Science, New York, USA, vol. 47 , no. 18 , pp. 47-66, June .

CloudTweaks, 2012, < http://www.Cloudtweaks.com/2012/04/Cloud-computing-types-of-Cloud-and-their-relevance-part-2/ >, viewed by *2* April 2012.

Breveglieri L., Koren I., & Maistri P. (2007), 'An Operation-Centered Approach to Fault Detection in Symmetric Cryptography Ciphers', *IEEE Transactions on Computers*, vol. 56, no. 5, pp. 635–649, MAY.

Ding Y., Neumann M. A., Gordon D., Riedel T., Miyaki T., Beigl M., Zhang W., & Zhang L.(2012), 'A Platform-as-a-Service for in-situ development of wireless sensor network applications ', in *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, Antwerp, Belgium, pp. 1–8, June 11-14.

Hassan S. A. Z. (2012), 'STAR: A proposed architecture for Cloud computing applications',in *2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)*, Dubai, United Arab Emirates, pp. 186–192.

Jadeja Y. & Modi K. (2012), 'Cloud computing-concepts, architecture and challenges ', in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, Kumaracoil, India, pp. 877–880, March 21-22.

Mazhelis O. & Tyrvainen P. (2011), 'Role of Data Communications in Hybrid Cloud Costs', *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on* , Oulu , Finland , pp. 138–145, August 30 - September 2.

Pan Y. (2011), 'Research on network database encryption technology', in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, Xi'an, China, pp. 690–693, May27-29.

Rivest R. L. , Shamir A. , & Adleman L. (1977) , 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems' , *Communications of the ACM*, vol. 21, no. 2, pp.120-126 , April.

Savu L. (2011), 'Cloud computing: Deployment models, delivery models, risks and research challenges', in *Computer and Management (CAMAN), 2011 International Conference on* , Wuhan, China , pp. 1–4, May 19-21.

Udin M. N., Halim S. A., Jayes M. I., & Kamarulhaili H. (2012) , 'Application of message embedding technique in ElGamal Elliptic Curve Cryptosystem', in *2012 International Conference on Statistics in Science, Business, and Engineering (ICSSBE)*, Langkawi, Kedah, Malaysia , pp. 1–6, September 10-12.

Umaparvathi M. & Varughese D. K. (2010), 'Evaluation of symmetric encryption algorithms for MANETs ', in *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, Coimbatore , Tamil Nadu, India, pp. 1–3 , December 28-29.

Wang H., Shea R., Wang F., & Liu J. (2012), 'On the impact of virtualization on Dropbox-like Cloud file storage/synchronization services' , in *Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*, Coimbra , Portugal , pp. 1 – 9, June 4-5.

Zou G., Zhang B., Zheng J., Li Y., & Ma J. (2012), 'MaaS: Model as a Service in Cloud Computing and Cyber-I Space ' , *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on* , Chengdu , China, pp. 1125–1130, October 27-29.

Wright D.(2007), <http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/wrighd/rsa_alg.html > , viewed by *6* April 2007.

Tawalbeh L., Mowafi M., & Aljoby W. (2012), 'The Potential of using Elliptic Curve Cryptography for Multimedia Encryption', *IET Information Security*, pp. 1-18, April 26.

Kulkarni G.,Gambhir J.,Patil T., & Dongare A.(2012),'A security aspects in Cloud computing' , *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on* , Beijing , China, pp. 547 - 550 , June 22-24.

Piper F. (1996),'BASIC PRINCIPLES OF *CRYPTOGRAPHY* ', *Public Uses of Cryptography., IEE Colloquium on* , London, United Kingdom, pp. 1-3, April 11.

Cisco.org, (2012). *Cisco CloudWatch Summer 2012***,** (On-Line), Available: **http://www.cisco.com/cisco/web/UK/assets/cisco_cloudwatch_2012_2606.pdf**

Kute V., Paradhi P., & Bamnote G. (2009), 'A SOFTWARE COMPARISON OF RSA AND ECC', *International Journal Of Computer Science And Applications*, Chikhli, India, vol. 2 , no. 1 , April / May.

Kak A. (2013), *'Lecture 14: Elliptic Curve Cryptography and Digital Rights Management'*, [Lecture] Computer and Network Security, Purdue University, USA, pp. 1-82, viewed by 26 February 2013, (On-Line), Available: <https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture14.pdf >.

Menezes A., Oorschot P., & Vanstone S. (1996), "Handbook of Applied Cryptography", CRC Press.

"Elliptic Curve Cryptography - An Implementation Tutorial - dkrypt.com." [Online]. Available: < http://www.dkrypt.com/home/ecc >. [Accessed: 06-Apr-2013].

Maurer U. M. & Massey J. L. (1993) , 'Cascade ciphers: The importance of being first' , *Journal of Cryptology*, vol. 6, no. 1, pp. 55–61.

Appendix

Appendix A: MATLAB

```
function [PlainText_Point] = ECCtext    %%Array='Elliptic_Curve_Cryptography';
fid=fopen('DS3.txt','r');   %% read dataset before encryption
Array=fscanf(fid,'%c');
fclose(fid);
fid1 = fopen('results.txt', 'wt');  %% file to write the results after encryption
tic
for l=1:1:length(Array)
[point] = Message_Encode(Array(l));
[Ciphered_point1 , Ciphered_point2] = Message_Encryption(point(1) , point(2));
fprintf(fid1,'%d','%d','%d ','%d
',Ciphered_point1(1),Ciphered_point1(2),Ciphered_point2(1),Ciphered_point2(2));
fprintf(fid,'%s\n ',' '); % ECC Decryption
[PlainText_Point] = Message_Decryption([Ciphered_point1(1)
Ciphered_point1(2)],[Ciphered_point2(1) Ciphered_point2(2)]);
Plaintext = Message_Decode(PlainText_Point(1));
Plaintext= char(Plaintext);   %%fprintf('the value after decryption is %d',Plaintext);
%[PlainText_Point] = Message_Decryption(Ciphered_point1,Ciphered_point2);
end
time=toc
time= time * 1000;
f=dir('DS3.txt');  %% read the description of file
s=f.bytes;          %% choose only the size in byte
fprintf('Elapsed Time in millisecond is %.f\n',time);
fprintf('Size of dataset in bytes before encryption is %f\n',s); %% size of raw file
f=dir('results.txt');
s=f.bytes;
fprintf('Size of dataset in bytes after encryption is %f\n',s); %% size of ciphered file
fclose(fid1);
 tic
 [Pk]= GET_Public_Key('A');
 Time_kG=toc;
 Time_kG= Time_kG * 1000;
 fprintf(' Time to generate pk in millisecond is %.f\n',Time_kG);
 end


function [Public_Key] = GET_Public_Key(Entity)
P=53330939;
A=2;
G =[503152 736];%[10000029 22528];% G point
if Entity == 'A'
```

```
Pr_k_A = 100 ;
Pu_k_A = Scalar_Multi(G(1),G(2),A,P,Pr_k_A);
Public_Key = Pu_k_A ;
 else
Pr_k_B =50 ;
Pu_k_B = Scalar_Multi(G(1),G(2),A,P,Pr_k_B);
Public_Key = Pu_k_B;
  end
    end

function [PlainText_Point] = ECC
I = imread('cameraman.tif');
I = im2double(I);
T = dctmtx(8);
 % Compression
dct = @(block_struct) T * block_struct.data * T';
%I=I(1:16,1:16);
B = blockproc(I,[8 8],dct);
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);
% ECC Encryption
S = size(I);
for i=1 :8: S(1)-1
for j=1 :8: S(2)-1
M=floor(B2(i,j));
B2(i,j)=B2(i,j)- M;
[point] = Message_Encode(M);
[Ciphered_point1 , Ciphered_point2] = Message_Encryption(point(1) , point(2));
B2(i,j+4)=Ciphered_point1(1);
B2(i,j+5)=Ciphered_point1(2);
B2(i,j+6)=Ciphered_point2(1);
B2(i,j+7)=Ciphered_point2(2);
end
end
PlainText_Point=1;
imshow(I), figure, imshow(B2)

% ECC Decryption
for i=1 :8: S(1)-1
for j=1 :8: S(2)-1
Ciphered_point1(1)=B2(i,j+4);
Ciphered_point1(2)=B2(i,j+5);
```

```
Ciphered_point2(1)=B2(i,j+6);
Ciphered_point2(2)=B2(i,j+7);
[PlainText_Point] = Message_Decryption([Ciphered_point1(1)
Ciphered_point1(2)],[Ciphered_point2(1) Ciphered_point2(2)]);
Plaintext = Message_Decode(PlainText_Point(1));
B2(i,j)=B2(i,j)+ Plaintext;
B2(i,j+4)=0;
B2(i,j+5)=0;
B2(i,j+6)=0;
B2(i,j+7)=0;
end
end
%Decompression
invdct = @(block_struct) T' * block_struct.data * T;
I2 = blockproc(B2,[8 8],invdct);
figure , imshow(I2)
%[PlainText_Point] = Message_Decryption(Ciphered_point1,Ciphered_point2);
end


function [x3 , y3, m] = ECADP(x1,y1,x2,y2,A,p)
% This function m-file performs Elliptic Curve addition over prime curves.
% Suppose we are working on the elliptic curve y^2 = x^3 + Ax + B
% Define P1 = (x1,y1)
% P2 = (x2,y2)
% Then P1 + P2 = P3 = (x3,y3) is defined by as below
% If one if the variables in infinity then we define P + infinity = P
% and the user should type in 'infinity' for both the x and y values
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if x1=='infinity'
x3=x2; y3=y2;
return
end
if x2=='infinity'
x3=x1; y3=y1;
return
end
if x1==x2
if y1==y2
if y1==0
display('P3 is infinity')
x3='infinty';, y3='infinity';
return
end
%m = sym( (3*(x1)^2 + A)/(2*(y1)) );
mnum = 3*(x1)^2 + A;
mden = 2*(y1);
m = mod( (mnum * inverse(mden,p)) , p );
% x3 = sym( m^2 - x1 - x2);
```

```
x3 = mod( (m^2 - x1 - x2) , p);
% y3 = sym( m*(x1 - x3) - y1 );
y3 = mod( (m*(x1 - x3) - y1) , p);
%x3 = [x3 y3 m];
return
end
display('P3 is infinity')
x3='infinty';
y3='infinity';
return
end
% m = sym( (y2-y1)/(x2-x1) );
mnum = y2 - y1;
mden = x2 - x1;
m = mod( (mnum * inverse(mden,p)) , p);
% x3 = sym( m^2 - x1 - x2 );
x3 = mod( (m^2 - x1 - x2) , p);
% y3 = sym( m*(x1 - x3) - y1 );
y3 = mod( (m*(x1 - x3) - y1) , p);
%x3 = [x3 y3 m];

function [M] = Message_Decode(pointx)
K =100000;
M = floor(pointx/K);
 End

function [PlainText_Point] = Message_Decryption(Cipher_point1 , Cipher_point2)
A=2;%2;
Pr_k_B =50 ;
G =[503152 736];%[10000029 22528];
P=53330939;
%Plain_point1 = Scalar_Multi(Cipher_point1(1),Cipher_point1(2),A,P,Pr_k_B);
%Plain_point1(2)= -Plain_point1(2)
v1=Cipher_point1(1);
v2=Cipher_point1(2);
for i = 1 : Pr_k_B-1 % 386 is the key
[v1 v2] = ECADP(v1,v2,G(1),G(2),A,P);
end
v2= -v2;
[x y] = ECADP(Cipher_point2(1),Cipher_point2(2),v1,v2,A,P);
PlainText_Point=[x y];
End

function [point] = Message_Encode(M)
K =100000; %% this not the security key, it is just a variable
P =53330939;
if M * K >= P
   disp('Error : M*K must be smaller than P\n');
else
```

```
for i = 0 : K-1
x = M *K + i ;
EC = mod(x^3 + 2*x + 7,P) ;
y = mod(sqrt(EC),P);
if abs(y) - round(y) == 0
    point = [x y];
    plaintext = floor(x/K);
%fprintf('The orginal Message is %d \n',plaintext)
   break;
end
end
if (i == K-1)
     disp('Error : M cannot be represented \n');
end
end
end

function [Ciphered_point1,Ciphered_point2]=Message_Encryption(x , y)
A=2;
P=53330939;
G =[503152 736];%[10000029 22528];% G point
K =20;%int8(100 * rand(1,1));
Ciphered_point1 = Scalar_Multi(G(1),G(2),A,P,K);% kG
Pu_B=GET_Public_Key('B');
 %C_point = Scalar_Multi(Pu_B(1),Pu_B(2),A,P,K)
v1 = Pu_B(1);
v2 = Pu_B(2);
for i = 1 : K-1 % 386 is the key
[v1 v2] = ECADP(v1,v2,G(1),G(2),A,P); % k Pb
end
% v1
% v2
[P1 P2] = ECADP(x,y,v1,v2,A,P);
Ciphered_point2=[P1 P2];
 end


function z=primetest(n);
% This function tests a number n for primeness.
% It uses the Miller-Rabin primality (compositeness) test
% z=1 means prime
% z=0 means composite
T=30;% hard coded amount of times we test
% increasing T decreases chance something is prime when it isn't.
% Probability of error is bounded by (1/4)^T
% First, express n-1 as n-1 = r * 2^s
if mod(n,2)==0,
   z=0;
   return;
```

```
else
r=n-1;
s=0;
while (mod(r,2)==0),
  s=s+1;
  r=r/2;
end;
for j=1:T,  %test T times
  a=2+floor(rand(1,1) * (n-4));
  y=powermod(a,r,n);
  if ((y~=1) & (y~=n-1)),
    k=1;
    while ( (k<=s-1) & (y ~= n-1)),
      y=mod(y*y,n);
      if y==1,
        z=0;  % n is composite
        return
      else
        k=k+1;
      end;  %end if-else
    end; %end while
    if (y ~= n-1),
      z=0;  % n is composite
      return
    end;
  end; %end if
end; %end for
 end; %end if
 z=1; %final case is it passed all tests and is prime

function y = randprime(N);
% This function finds a random prime between 1 and N
% The prime is tested using Miller-Rabin
N1=N-1;
flag=1;
while flag,
  y=1+floor((N1)*rand(1,1));
  if primetest(y),
    return;
  end;
end; %end while
%%%%%%%%%% scalar Multiplication %%%%%%%%%
function [point] = Scalar_Multi(x,y,A,P,k)
[v1 v2] = ECADP(x,y,x,y,A,P);
for i = 2 : k-1 % 386 is the key
[v1 v2] = ECADP(v1,v2,x,y,A,P);
end
point = [v1 v2];
end
```