KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

KNUST

AN ENHANCED ASYMMETRIC CRYPTOSYSTEM BASED ON MULTIPLE KEY SYSTEMS

BY

OKYERE-GYAMFI STEVE

(B.Sc. COMPUTER SCIENCE)

AN ENHANCED ASYMMETRIC CRYPTOSYSTEM BASED ON MULTIPLE KEY SYSTEMS

KNUST

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI GHANA

BY

OKYERE-GYAMFI STEVE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF PHILOSOPHY DEGREE IN COMPUTER SCIENCE AT THE DEPARTMENT OF COMPUTER SCIENCE, COLLEGE OF SCIENCE, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI GHANA

MARCH, 2018

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma at Kwame Nkrumah University of Science and Technology, Kumasi or any other educational institution, except where due acknowledgement is made in the thesis.

Okyere-Gyamfi Steve …….. …….

Signature Date

(PG4723115)

Certified by:

Dr.J.B Hayfron-Acquah …….. …….

Supervisor Signature Date

Certified by:

Dr. Michael Asante …….. …….

Head of Department Signature Date

KNUST

ABSTRACT

An increase in network technology development has its own downside; thus as more connections are established with various global computer networks daily, the more exposed the connected systems are to unauthorized access, thus making security of data very important to address. Internet based transaction applications such as internet banking, online shopping etc involves sharing of very sensitive information between two or more parties that should be confidential. This requires very secure end-to-end connections that will ensure the data integrity, confidentiality, authenticity, etc. Cryptography is one of the most reliable and best, if not the best way to keep sensitive data from unauthorized users. This implies a good cryptosystem that maximizes security of the data/information been transferred and minimizes a substantial amount of delay time is needed. This is dependent on the particular cryptosystem one chooses to secure information. Also of the two known types of cryptosystems, the best in security is asymmetric cryptography, which uses two different keys; one for encryption and the other key for decryption, whiles symmetric cryptosystems use the same key for both encryption and decryption. The essential features of asymmetric cryptosystems that determines a cryptosystems efficiency and security are; encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput, throughput, randomness, key length, Operation per Instruction (O/I).This research seeks to examine these properties of some asymmetric cryptosystems and subsequently develop a proposed one that is more secured and efficient. The results of this research clearly demonstrate that, the proposed cryptosystem has better results for all the properties stated above.

KNUST

ACKNOWLEDGEMENTS

I express my sincere gratitude to the almighty God for his grace to do this work. And to my supervisor Dr. J.B. Hayfron-Acquah, for his support and guidance, I am very honored to study under your feet. I would like to also acknowledge all my lecturers of the computer science department who contributed indirectly or directly to this work. I am also thankful to cryptosystem researchers whose work inspired me to undertake this research.

Finally I would like to thank my parents, Mr. and Mrs. Okyere-Gyamfi who always want the best for me for their love and encouragement. I really appreciate your sacrifice and determination. And to my dearest wife Vivian Okyere-Gyamfi, I say God richly bless you.

NYANSAPƆ W ƆSANE NO BADWENMA

DEDICATION

This thesis is dedicated to my sons Emmanuel Joshua and Joseph Benjamin Okyere-Gyamfi.

KNUST

TABLE OF CONTENT

PAGES

LIST OF FIGURES

KNUST
NYANSAPƆ WƆ SANE NO BADWENMA

CHAPTER ONE

1.1 Introduction

The ongoing unstoppable trend/effort by various IT stakeholders towards increasing computing power has equally caused the network computing world to progress at an extraordinary pace, such that network properties like connectivity, speed, etc is estimated to double every year, thereby resulting in an exceedingly amazing increase in the number of users connected to the network computing world each year compared to previous years summed up together.

This rapid and continual change and development in network technology is also transforming our world and various aspects of our daily life, like business life, legal life, social life, etc.

However this increase in network technology development has its own downside; the more the connections established with various global computer networks daily, the more exposed the connected systems are to unauthorized access.

This is because common practices like network-scanning, spoofing, etc have escalated, thereby making information sharing risky.

Moreover, most recently, the emergence of internet based transaction applications such as internet banking, online shopping and bill payment, etc which involves sharing of very sensitive information between two or more parties, require very secure end-to-end

connections that will ensure the data/information's integrity, confidentiality, authenticity, etc, most popularly called *CIA triad* (Gambhir and Ankit, 2014).

Thus making security a very important element in network technology development that needs to be addressed, because it is by it that services and data / information are protected from destruction/change and unauthorized intrusion.

Security in computing is the protection of automated information system/information to meet the objectives of availability, confidentiality and preserving the integrity of information system resources such as software, hardware, firmware, telecommunications and information or data.

This problem of security has brought about the demand for the development of various techniques or technologies such as passwords, biometrics, patterns, cryptography, etc to help eliminate network security issues especially keeping data/information from unauthorized access.

But of all these techniques, some have some shortcomings which make them unreliable; example passwords with short ranges are easy to be cracked using simple guessing and/or brute force attack. Also, the biometric gadgets are individualizing in nature (Essays U.K.com), they can fail in recognition of individuals (Elliot S.J. et al, 2004) and may also produce harmful effects on the user and moreover they are often times costly (BiometricSecurity-Devices.com). But of all these techniques/methods the one that proves to be the best solution for all the shortcomings stated above and also more reliable in keeping sensitive data/information like bank account details, etc confidential from unauthorized users is cryptography (Yogita, 2016).

This implies, a good cryptosystem is needed, thus one that maximizes security of the data/information being transferred and minimizes a substantial amount of delay time.

Cryptography which is one of the main methods used in network security is the science and art of manipulating or altering information in order to make it understandable to authorized persons by use of certain processes/methods (i.e. algorithms) in order to protect the confidentiality, integrity, authenticity etc of the information. This can be accomplished by using a major method in cryptography called encryption which is a major method used to secure sensitive data by turning it into scrambled information. In securing data using this process, processing time or return time and security are very important resources to consider. This is dependent on the particular cryptosystem one chooses to secure information or data.

1.2 Overview of Cryptography

Cryptography is the study and practice of techniques that is used to make communication secure from intruders like hackers and attackers over networks. It deals with the protection of information by altering the information into cipher text to make it useful only to the intended party who can convert it back into plaintext. Cryptography makes the message unintelligible to a third party or an outsider by various ways of transformations. This practice is performed by *cryptographers* (Sheela and Prakash, *2013*).

In computing, the function of cryptography is to transform (encrypt by using a key or no key) information for authenticity or secrecy purposes. These are achieved with strong

algorithms and good key management techniques (for those cryptosystems that use keys) (Khalique *et al*, 2010).

Some of the main goals that cryptography seeks to achieve are;

i. Privacy or confidentiality: This service help to keep information content secret from unauthorized parties. The terms privacy and confidentiality are synonymous. Confidentiality can be provided in different ways, but cryptography deals with protecting data by the use of algorithms to render information unintelligible. ii. Data Integrity: It refers to the ability of an information/data to remain intact or true or unchanged even if manipulations like deletion, insertions, substitutions, etc are performed on that information/data by unauthorized parties/persons. The integrity of data helps to ensure that the data/information is not tempered with or manipulated, by unauthorized persons and that there is detection if they should try to temper with it.

iii. Authentication: This service refers to identification of an information/data. Thus it deals with both the origin of a specific information/data and its entity authentication. This implies the ability of two or more parties communicating to be able to identify and authenticate each other and also to be able to authenticate or trace the information(s)/data delivered to each other to their origin, content, time sent etc.

iv. Non-repudiation: This is also a major goal that cryptography seeks to achieve and it involves the prevention of communicating entities from denying their previous

actions, commitments, status etc. When there arise disputes that an entity denies certain actions taken by him or her, there need a means to resolve the situation

v. Access Control:Prevent unauthorized users access to data(Delfs and Knebl, 2006).

There is a basic model upon which every cryptosystem is based on, which is called the basic cryptographic model in Figure 1.1. There exist two main components in cryptography and these are Encryption and Decryption algorithms; and a third component called Key (thus for cryptosystems that use a key in both encryption and decryption).

Below in Figure 1.1 is a model of the basic cryptographic model.

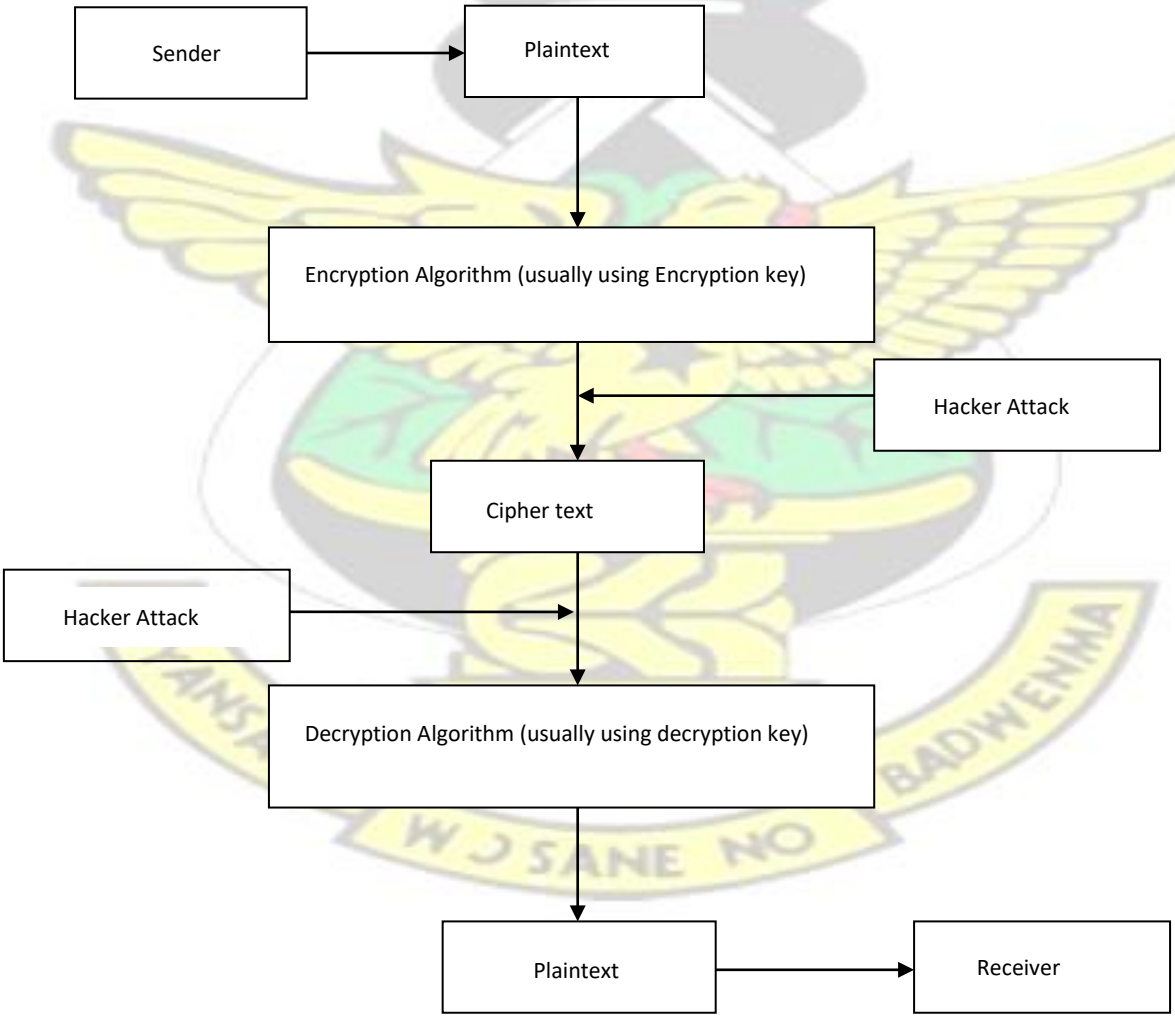


Figure 1.1 Basic cryptographic model

One effective mechanism everywhere that is used to achieve the security of data is encryption which is a part of the first part of the basic cryptographic model. It is a process of converting data into a coded form (usually by the use of a key usually referred to as the public key) (Stallings, 2003). It deals with concealing the content in a message in a way that the content of the original message can be recovered or retrieved back through a decryption process. Encryption is aimed at preventing access to crucial information by unauthorized entities. It occurs when information passes through techniques such as table references, shifting technique or mathematical operations by using an encryption algorithm that performs operations on the original data or message (plain text) to convert it to cipher text at the sender end which can only be useful or read when decrypted at the receiver end by the use of a decryption algorithm to convert the cipher text to plaintext (Khalique and Bansal, 2010).

To encrypt a message, an encryption algorithm is used. For cryptosystems that use keys, an encryption key is also needed and the nature of the key is dependent on the algorithm used. Encryption does not prevent interference, but help to deny the content of the message to the interceptor. Below in Figure 1.2 is a model of the encryption process.

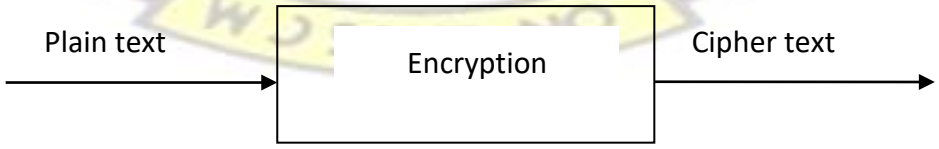


Figure 1.2 Encryption process

Decryption is also a very vital component in the basic cryptographic model. This is because the decryption process helps to convert cipher text to plaintext by the use of a decryption algorithm that performs operations on the cipher text (scrambled message) in order to get back or recover back the plain text. The cryptosystems that use keys use a decryption key (usually referred to as the private key or secret key) for decryption of data (Khalique and Bansal, 2010). Only authorized/intended recipient can decrypt the message easily (Stallings, 2003). Below in Figure 1.3 is a model of the decryption process.

Figure 1.3 Decryption process

The third part of the basic cryptography model (which pertains to some cryptosystems) called Key refers to a value or word used for encryption of the plaintext and/or the decryption of the cipher text.

For asymmetric cryptosystems the key is made up of two parts, thus the encryption key/public key which is for encryption, and the decryption key or private/secret key which is used for decryption. However for symmetric cryptosystems there is only one common key for both encryption and decryption and that key is secret.

Multiple keys can be used to perform encryption and/or decryption.

The Key Size is measured in bits. The larger the key size, the more secure the information communicated (Alqdah and Hui, 2008).

In the absence of cryptography information sharing or storing with/without computing networks can be devastating. Thus the uses of cryptosystems cannot be ignored and are inevitable. Below are some various uses of cryptography in this modern time;

Secrecy in transmission is one major importance or benefit of cryptosystems. The prevention of data to be read by a third party is a main goal of cryptography. For example for key based cryptosystems like symmetric and asymmetric cryptography the private/decryption key must be secured from access by unauthorized parties because any party who get access of the private key can decrypt the encrypted message. Thus only an intended receiver can decrypt an encrypted message that has been transferred to him or her thus, making the information remain secret and secure (Oded, 2004).

Another major use of cryptosystems is for secrecy in storage. Applying cryptographic techniques on message or data in storage is referred to as storage encryption. The secrecy in data storage can be achieved and maintained by encrypting the data in storage. Thus for key based cryptosystems, information cannot be accessed without providing a key thus preventing the unauthorized access and misuse of information on a storage device when stolen (Mihir, 2000). Cryptography is therefore very important in securing both data on transit and on storage.

The integrity of data during transmission is crucial and can be preserved by the use of cryptography, which prevents data being transferred from been altered. For the prevention of intentional or accidental modification of message that can cause erroneous actions during transmission, techniques of cryptography are employed. Also it is important to

maintain integrity during electronic funds transfer, since millions of money can be lost by a bank if a transaction done without the use of cryptography is intercepted (Mihir, 2000).

Moreover the integrity of stored data is ensured by the use of cryptographic based access control systems such as keys, locks and other techniques for the prevention of unauthorized parties from getting access to data. To get the validity of stored data, cryptographic checksums are used (Oded, 2004). Thus making integrity of transmitted data another major area where cryptosystems can be used.

Also another major use of cryptography is for the verification of a user to find out whether he or she has authority to access data which is known as authentication. The use of cryptography in data authentication during and after its transmission is very vital, example some cryptosystems are used in the provision of passwords for identity authentication. Therefore for a more efficient and reliable identity authentication, cryptography is used.

Moreover another major use of cryptography is in creating digital signatures which is a mechanism whereby a message is authenticated by proving that a particular sender sent a message just like signature on a paper. Digital signatures are of need if the parties performing the transaction are not close physically or there is a big business deal. This can be done by the use of hashing and public key cryptosystems (Oded, 2004).

Cryptosystems are also used in Credentialing systems, Electronic money, Threshold cryptosystem, secure multi-party computation etc (Mihir, 2000).

1.3 Key-Based Cryptography

The presence/absence and/or type/number of keys determine the type of cryptosystem and also the encryption phase and decryption phase involved in that cryptosystem. Depending on the number and kind of key(s) used, key based cryptosystems are classified or grouped in two; the first is Symmetric key cryptosystems (usually called private/secret/conventional key) and the second is Asymmetric key cryptosystems (usually called public-key cryptosystems) (Gagandeepshahi and Charanjitsingh, 2013). Below in Figure 1.4 is a diagram of the classification of key based cryptography.

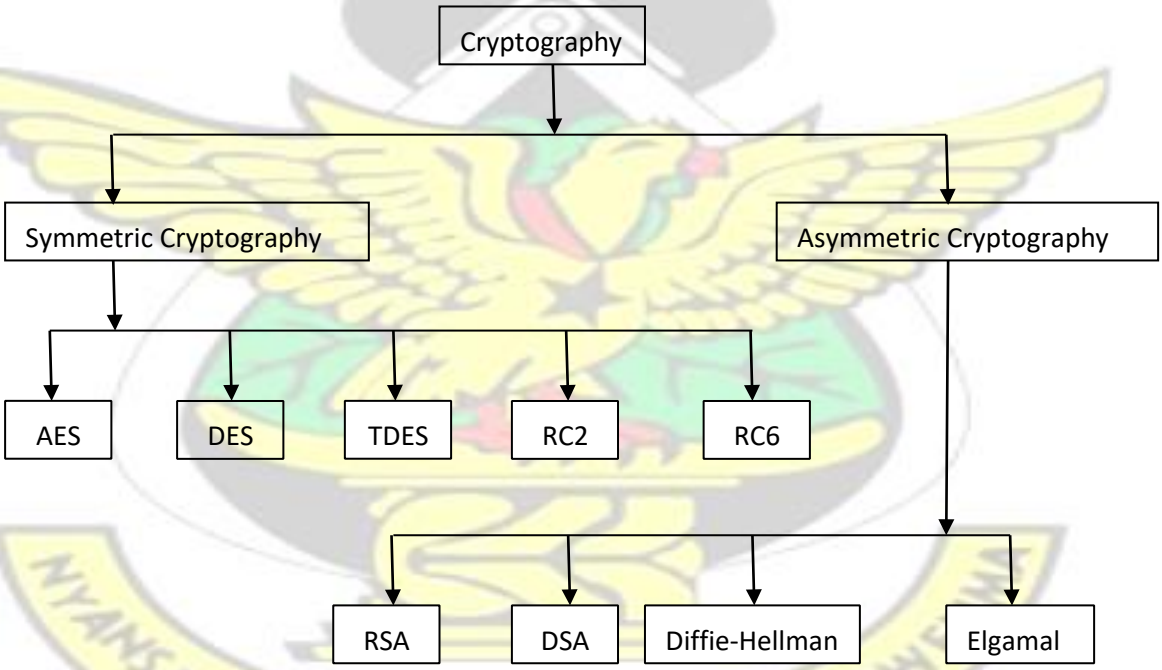


Figure 1.4 Classification of key based cryptography/cryptosystems

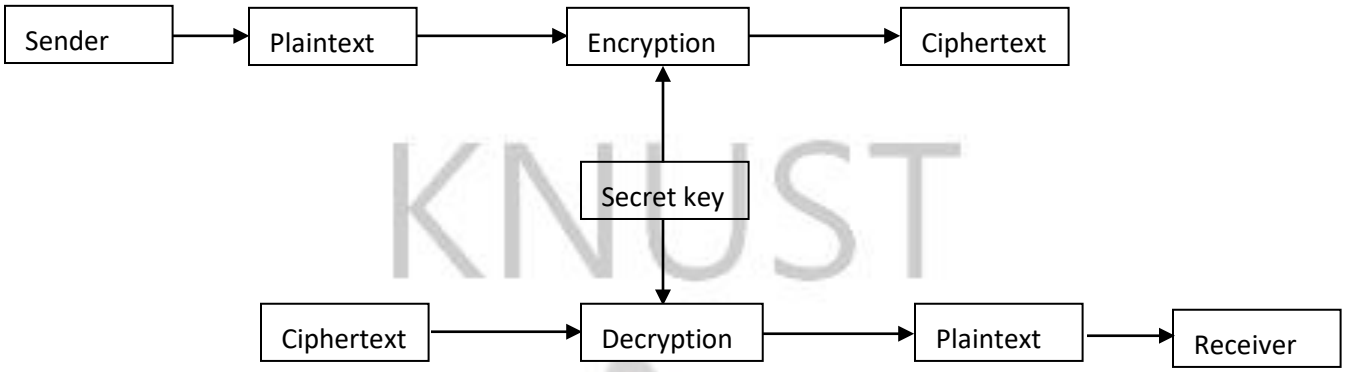


Figure 1.5 Symmetric key cryptography process

In Symmetric cryptography, the receiver and sender in this cryptosystem apply or use a common key (called private/secret key) in encrypting and decrypting messages. Usually both receiver and sender of the message agree in secret upon this common key they want to use. Only authorized people thus both receiver and sender of the message know of it, and this accounts for the name private key cryptography. The private key should be distributed in a very confidential mode (Bhagat *et al*, 2013). In symmetric cryptography, the encrypted data is communicated without the attachment of a public key.

There are two types of symmetric cryptosystem. These are stream ciphers; they encrypt messages as they are streamed thus one at a time. And the second type is block ciphers; they encrypt messages (in bytes) in a single unit. The plaintext is padded so that it becomes a multiple of the block size. 64 bits of blocks were commonly used (Gagandeepshahi and Charanjitsingh, 2013).

Some common symmetric cryptosystems are Data Encryption Standard (DES), Triple Data Encryption Standard (TDES), Advanced Encryption Standard (AES), RC2, RC6 etc. (Bhagat *et al*, 2013). Below in Figure 1.6 is a model of the asymmetric key cryptography process.

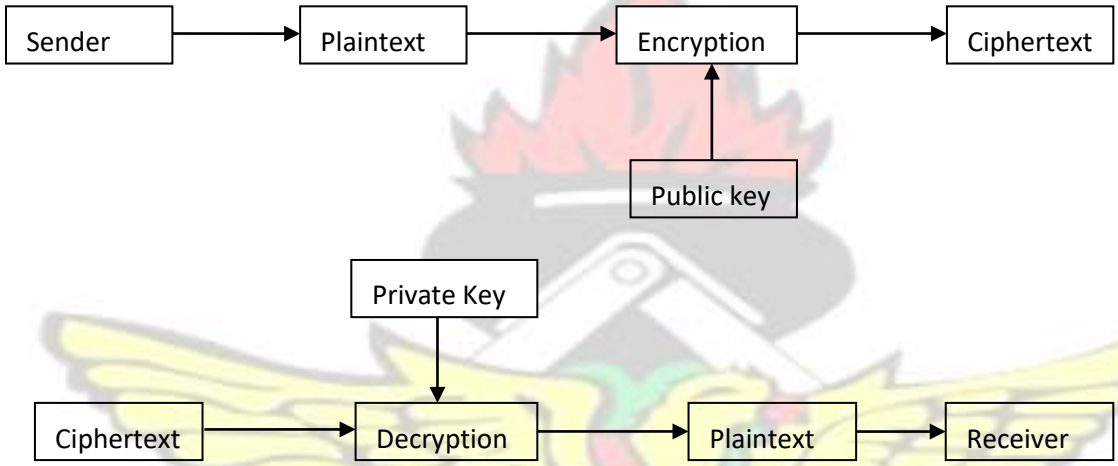


Figure 1.6 Asymmetric key cryptography process

The constant need to communicate sensitive information over different global networks and subsequent increase in tempering of computer networks by attackers brought about the need for the use of symmetric cryptography on a large scale. But due to its shortcoming in key distribution researchers and cryptographers had to come out with other models of key based cryptography other than symmetric cryptosystems.

Thus, symmetric cryptosystems were found to have a big loophole, in that when an attacker gets hold of the secret key used for encryption, that attacker can use the same secret key to decrypt the encrypted message. This challenge gave way to the asymmetric/ public key cryptosystems (Boneh and Franklin, 2001).

In Asymmetric cryptography, the sender and receiver in this cryptosystem apply two different keys for encryption (convert plaintext to encoded/scrambled message/cipher text) and decryption (convert cipher text to plaintext) of message. In asymmetric cryptography, each workstation or user that takes part in a particular communication have two different keys namely, the public/encryption key and the private/decryption key and these keys are used to perform cryptographic operations. The key for encryption is made available to all who would like to send an encrypted message to the party with the secret key and therefore the name public key; and the key for decryption is stored in secret. It is not possible to derive easily a corresponding secret key from a particular public key. A particular device or user alone know the secret key(that is why it is called private/secret key) but the public key is shared to every authorized device(s) or user(s) who partakes in the inter-networks communication. A communicating party can easily exchange the public key online because the knowledge of it does not cause the security of the asymmetric cryptosystem to be compromised (Gennaro, 2000).

Examples of common asymmetric cryptosystems include; Diffe-Hellman, RSA, ElGamal, Elliptic Curve cryptography etc. and examples of protocols that use these cryptosystems include; SSH, SSL, Digital Signature Standard (DSS) (Millett and Holden,

2003).

Asymmetric Cryptosystems are known to have a number of advantages. Notable among them are;

i. Distribution of key problem is eliminated because there is no need to exchange keys.

ii. There is increase in security because private keys do not need to be revealed or transmitted to anyone.

iii. The only key that must be kept secret is the private key. This guarantees the authenticity of public keys. (Delfs and Knebl, 2006).

iv. Key pairs can remain unchanged depending on how it is used for a period of time or for many sessions.

v. Total number of keys needed in large network can be smaller than the one in a Symmetric cryptosystem scenario (Kahate, 2003).

But asymmetric cryptosystems also have some disadvantages. Notable among them are;

i. They are slower than symmetric cryptosystem.

ii. Keys in this cryptosystems are much longer and computationally costly than the ones in symmetric cryptosystems (Delfs and Knebl, 2006).

iii. Higher processing power for message encryption and decryption. (Kahate, 2003).

iv. They may be also acutely prone to vulnerability attacks by unauthorized third parties like hackers.

KNUST

1.4 Research field and subject of the study

The subject under study is asymmetric cryptosystems and the research field is computer science. Cryptography is making data unintelligible/unreadable by scrambling the content mostly for secure transmission. Asymmetric key algorithms/cryptosystems which is the most secure type/model of key based cryptography, use two different keys for encrypting (using a key made public) and decrypting data (using a key kept as secret). This is a very interesting subject area that needs constant and improved probing and analysis in order to find more improved techniques and methods to make asymmetric cryptography more efficient with the help of modern technology and ideas. But not much research has been done in this subject of study (with most not detailed enough), and even recent advances in computing, network and communication power and technology are making those researches which have been made obsolete. Therefore, analysis of existing asymmetric cryptosystems will be done in this research and a more secure and efficient asymmetric cryptosystem compared to the existing ones will be developed and comparatively analyzed with the existing ones based on modern ideas and technology.

1.5 Problem statement

Data security issues have escalated in this information age, due to the rise in information sharing and transfer over open channels or networks which are vulnerable to interception. Many researchers and computer scientists have sought to solve this security "puzzel" by developing various data security techniques such as passwords, symmetric and asymmetric

cryptography etc, of which the most acclaimed as best in data security is asymmetric cryptography.

Yet still there remains, some minute problems of security and some major issues with processing/return time for most asymmetric cryptosystems. Thus most of the existing asymmetric cryptosystems have a problem with either security or efficiency but mostly with efficiency. Moreover comparative analyses of asymmetric cryptosystems are few. And even those that have been done are based on only security or a few properties, although security and processing time of asymmetric cryptosystems are important resources to consider when one is to choose an asymmetric cryptosystem to secure data.

This makes it necessary to analyze various common existing asymmetric cryptosystem and develop an efficient and more secured asymmetric cryptosystem so that users will choose suitable asymmetric encryption algorithm to secure their data.

1.6 Objectives

From observations it is concluded that there is a need to develop asymmetric cryptographic model for data security. The following objectives will be addressed by my study:

i. To analyze and study the various asymmetric cryptosystems used for data security and their various strengths and weaknesses.

ii. Develop a more secure and efficient asymmetric cryptographic model compared to the existing ones and compare the proposed approach with existing ones.

iii. Create awareness of the benefit of knowing and selecting an appropriate asymmetric cryptosystem.

1.7 Research questions

i. What benefit does knowing the characteristics, strengths and weaknesses of various asymmetric cryptosystems, provide to the user and system in securing /protecting data.

ii. What will be the importance in developing a new secure and efficient asymmetric cryptosystem compared to existing ones?

iii. What benefit does knowing and selecting an appropriate asymmetric cryptosystem provide to the user and system?

1.8 Significance of the study

The principal goal of the study is to build an asymmetric cryptosystem that will be more efficient, provide higher security than the various asymmetric cryptosystems and overcome their security vulnerabilities.

1.9 Organization of thesis

This thesis is organized into five chapters. They are chapter one (introduction) which shows the overview of cryptosystems and gives one a fair idea of the basis of the research. The chapter two (literature review) reviews prior/previous thesis and studies done by different researchers who have worked in the area of asymmetric cryptography. The third chapter (methodology) looks at the experiments and data to be used in the study. The chapter four (discussions and results) includes comparison between proposed approach and other

algorithms. Finally, the fifth chapter (conclusions and recommendations) presents the summaries, conclusions and recommendations of the study and directions on choosing particular cryptosystems for data security.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

Data security for data in transit or in storage is an afterthought for most PC users but it is a major issue and priority for companies or businesses of any size. Breaching security costs companies millions each year. Even if a company will not lose much, protection of information must be of a high priority. Threats that a user needs to deal with has increased due to new technologies emerging and the growth of information technology power. No matter what kind of information it is, whether social security numbers, bank accounts or simple telephone number, it is important it is known only by the authorized or intended persons (normally the sender and receiver). Several security technologies exist, but asymmetric cryptosystems/public key encryption is a technology that everyday computer users must know about, since they are used for the secure transmission of secret keys, sensitive and confidential data without the communicating parties agreeing on a key in private. This chapter deals with a study of various common asymmetric/public key cryptosystems and how they work, their strengths and weaknesses.

Some popular asymmetric cryptosystems developed are RSA (which is named after their developers; Rivest, Shamir and Aldeman), Diffie-Hellman(also named after its developers Diffie and Hellman), Elgamal(also named after its developer Elgamal) and Elliptic Curve Cryptography(ECC) Cryptosystems, etc. Some of these asymmetric cryptosystems also have their variants that have also been produced. Thus others have tried to improve upon these original ideas in asymmetric cryptography, especially RSA, Diffie-Hellman, Elgamal and ECC cryptosystems. Nonetheless, all the major or pioneer asymmetric cryptosystems and their variants can be grouped into three categories based on their design, namely; Integer Factorization model, Discrete Logarithm model and Elliptic Curve model cryptosystems. The Integer Factorization model cryptosystems like RSA are based on the principle or assumption that it is hard to factorize a large integer whiles the Discrete Logarithm model cryptosystems are based on the principle of how hard it is for discrete logarithms to be computed whiles finally Elliptic Curve model cryptosystems are based on elliptic curve equation in mathematics. But of these three categories of asymmetric cryptosystems, the most popular and most acclaimed to be secure is the Integer Factorization model cryptosystems because it is harder to factorize compared to the discrete logarithm model cryptosystems and the others (Orman and Hoffman, 2004). Below is a discussion on all these three categories of asymmetric cryptosystem models, the way they work, their strengths and weaknesses.

Figure 2.1 shows the different models of asymmetric cryptosystem and examples.

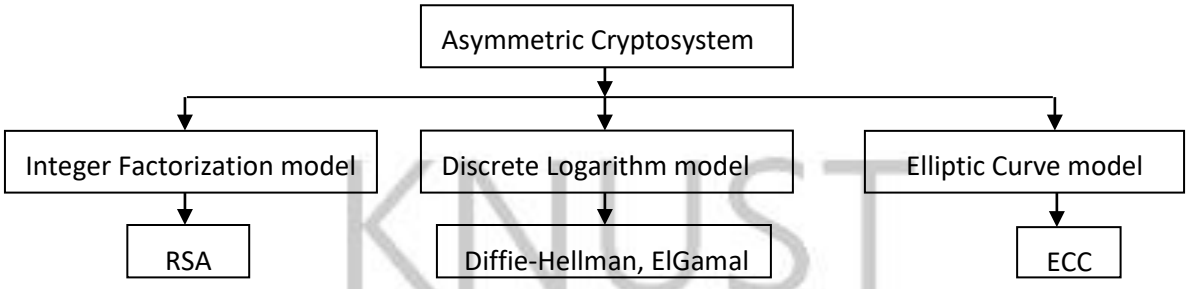


Figure 2.1 Classification of Asymmetric Cryptosystem

2.1 Integer Factorization Cryptosystem Models

Rivest *et al (*1978*)* designed RSA cryptosystem at Massachusetts Institute of Technology (MIT) (Kakkar *et al*, 2012). This algorithm can be divided into three stages namely; key generation, encryption and decryption. The RSA cryptosystem make use of two keys, that is; public key and private key. These keys are for encryption and decryption of data respectively. Messages are encrypted by the use of the public key of the receiver, who then uses the secret key to decrypt the message. Two random prime numbers are used to generate public and secret keys.

For the RSA Key Generation Algorithm;

i. Two large primes *p* and *q* that are random and are approximately of equal size are generated so that their product n = p*q is of the required bit length.

ii. Compute the value n = p*q and (φ) phi = (p-1)(q-1) iii. Choose a particular integer *e*, 1 < e < phi, such that gcd (e, phi) = 1.

iv. Compute the exponent *d*, such that 1 < d < phi, and that ed ≡ 1 (mod phi).

v. The public key then is (n, e) and private key is (n, d). The values d, p, q and phi are kept secret.

vi.The value n is called the modulus. vii.The value e is called the public/encryption exponent.

viii.The value d is called the secret/decryption exponent. ix.The key pair (n, e) forms the public key and it is made available to all or public.

x.The key pair (n, d) forms the private/secret key and it is kept private (Sharma *et al*, 2012).

Encryption in RSA is made possible by using the public key to generate a cipher text from plaintext.

The encryption steps are as follows;

i.The sender gets the recipients public key.

ii.The message (M) or plaintext is first of all converted to a positive integer.

iii.The cipher text (C) is then computed. iv.The cipher text is then sent by the sender

v.This is mathematically represented as: $C = M^e \bmod n$

Decryption in RSA is done by the use of the private key to extract plaintext from cipher text.

The decryption steps are as follows;

i.The private key is used for this process.

ii. Extracts plain text from cipher text which is represented as an integer.

iii.This is mathematically represented as: $= C^d \bmod n$ (Kim and Lee, 2004)

After the first RSA algorithm, a lot of researchers have also tried to produce variants of it or suggested ideas that will help improve RSA algorithm.

One of such is (Williams, 1980) made a modification to RSA public-key cryptosystem. His suggestion was that, if the procedure of encryption was divided into some number of operations then after some little more operations the use of the remainder/modulus can be factored. This cryptosystem appeared similar to RSA. However, the limitation/setback of this cryptosystem was the use of very large prime numbers which in turn brought about mathematical errors.

Again (Sun *et al,* 2007*)* made a proposal of a dual RSA cryptosystem and also made analysis of the security of the cryptosystem. These researchers presented new RSA variant cryptosystems whose algorithm for key generation produces a distinct pair of RSA keys which have similar private and public exponents. Authentication and blind signatures are two applications for Dual RSA. There was comparison of the security of the Dual RSA and RSA with the use of small values of "e" and "d" and one main limitation of the use of dual RSA is an increase in complexity in terms of the computation involved in the key generation.

Moreover, (Ayele *et al,* 2013) created a modified RSA which used two public keys instead of one which are sent separately and is more secure and less susceptible to brute force attack than the original RSA. One major setback of this modified RSA is that it has high communication overload and this is due to the fact that the there are a pair of public keys and the pair are sent separately.

A major strength or backbone of the RSA cryptosystem is the assumption or principle that it is difficult to factorize out two large prime numbers from a single large integer which is the modulus (Orman and Hoffman, 2004).

Over the years researchers have tried to check if there is a flaw in RSA, but all proved futile. But recently on Valentine's Day 2012, in a publication titled "Ron is Wrong, Whit is Right" (Lenstra *et al*, 2012) a group of European and American researchers (i.e. mathematicians and cryptographers) conducted a comprehensive survey of public keys which were collected with authorization from the world wide web (www) and one of their main focus or target was on RSA cryptosystem. Thus they did this research to test the claim by RSA Security Company of strength, secrecy, randomness and distinctness of

RSA public keys and also of other public keys algorithms based on Diffie-Helman like Elgamal and DSA. For RSA, (Lenstra *et al*, 2012) came to realize that out of every thousand keys two may be vulnerable or not secure. Thus after collecting and examining millions of openly accessible or handy public keys as they can and employing the Euclidean algorithm (an efficient technique which is used to compute the greatest/largest common divisor (gcd) of two integers efficiently and quickly) to examine those public keys (which is in the form of numbers), they realized that a very small percentage (about 1.1%) of these numbers were not actually random. This test or analysis of public keys was done several times on about three (3) different sets of millions of public keys. Thus the result of this test or analysis showed that it is viable to find the private/secret key or underlying numbers that was used to generate the public key.

RSA Company subsequently responded to this flaw finding of Lenstra *et al* (2012) concerning the RSA algorithm in a publication in the IEEE Spectrum written by (Moore, 2012). RSA responded that this flaw found by Lenstra *et al* (2012) in RSA cryptosystem,

is not from the algorithm itself but from its implementation. Thus according to RSA Security Company the supposed RSA cryptosystem flaw based on Lenstra *et al* (2012) survey which showed that some of the randomly generated prime numbers used to generate the keys are duplicates, should be attributed to some of the hardware (i.e. embedded systems) and software used to generate the random numbers, thus they are not accurate and efficient enough in doing the distinct random number generation. Thus some or most of the random number generation software and embedded systems end up producing certain duplicate random numbers that were found by the researchers. Moreover surprisingly, some Intel engineers (Taylor and Cox, 2011) and other engineers wrote in IEEE Spectrum of how they are finding ways to massively improve upon the randomness of encryption numbers, thus by using digital circuits on a processor of a computer or through changes in transistor characteristics in RFID memory chips.

Moreover on Dan Kaminsky's Internet Blog on Valentine's Day 2012, he responded to the same paper on the flaw of RSA cryptosystem; he attested to the fact that the paper was good in its survey but its thesis was wrongly done and was also biased. In a statement he wrote, he said that because there was some number of vulnerable RSA keys in the (Lenstra *et al*, 2012) survey, doesn't mean RSA cryptosystem is in itself all together vulnerable or that it was a fault of RSA cryptosystem (thus it caused those keys to be vulnerable) or there is a flaw in RSA cryptosystem. Thus there is no correlation between the vulnerable keys and RSA cryptosystem. He pointed out that more than fifty percent of the keys examined in the survey are RSA keys whiles the rest are those of other asymmetric cryptosystems like ElGamal etc. Therefore if the other asymmetric cryptosystems like ElGamal with

fewer keys analysed, were still able to have vulnerable keys then it means RSA have an advantage over the rest. Thus he concluded that the thesis is wrong although the survey is good. However for smaller values of the randomly generated prime numbers, the keys that are generated from it becomes weak, if the values are too large too, more time is required during the encryption and decryption processes and degrades performance. In addition to this, according to (Sun *et al*, 2007), it was revealed that, in RSA cryptosystem; the disadvantage is a low encryption speed. These responses to the thesis on RSA flaw found (Lenstra *et al*, 2012) and all the other literature on RSA and its variants clearly shows that RSA cryptosystem is a little vulnerable but if an improved RSA cryptosystem with a bigger modulus which is neither too small or extremely large is used it will be difficult to use the factorization method or number field sieve (Lenstra and Lenstra, 1993) to determine the factors and thereby get the private key. Also that improved RSA cryptosystem with good key management will be more secure, efficient and reliable cryptosystem and also if the implementation of random number generation is done well; it then ultimately becomes an utterly flawless cryptosystem.

2.2 Discrete Logarithm Cryptosystem Models

Another major public key cryptosystem is the Diffie–Hellman cryptosystem which was published by Diffie and Hellman (1976) and was based on the difficulty of computing the discrete logarithm theory. It is designed to allow two parties who do not know one another to collaboratively create a common secret/private key over communication channels with no or little security. Thus for subsequent communication between the two parties to be

encrypted a private key which has been created is used with the use of a symmetric key cipher (Arya *et al*, 2015).

Assuming the two parties K and L want to establish or settle on the encryption/ decryption keys to be shared among themselves, then this is how the key exchange algorithm of Diffie-Hellman would work (Arya *et al*, 2015);

i. First of all, the parties K and L would consent on two big prime numbers r and s, and they do not have to be secret. K and L can use their own private channel which may not be secure to communicate and agree on them.

ii. K then selects a different large random integer j and computes b such that b=s*j mod r. iii. K communicates the computed resulting number b to L.

iv. L also independently selects a different large random integer h and computes f such that f=s*h mod r.

v. L also communicates the computed resulting number f to K. vi. K then calculates the secret key Q1 as follows Q1= f*j mod r.

vii. L also calculates the secret key Q2 as follows Q2=b*h mod r.

The strength of this cryptosystem is based on how cumbersome it is to compute discrete logarithms. Also, the private key is never actually transmitted on the communication channel and also it can be transmitted offline between the two parties. But the limitations of Diffie-Hellman cryptosystems is numerous; e.g. usually it is for only key exchange (Arya *et al*, 2015) and also according to (Andrian *et al*, 2015) it has limitations like use of standard prime groups which are not properly generated, precomputation of primes, etc. ElGamal cryptosystem is also based on discrete logarithm theory and a variant of Diffie–

Hellman. Strength of ElGamal cryptosystem is based on the similar idea that it is impossible to calculate or factorize discrete logarithms within a pragmatical or realistic amount of time given large prime number and also the simplicity involved in multiplying the symmetric key by the message, thus public key creation (Singh and Kumar, 2012). A user (e.g. Kofi) of ElGamal cryptosystem must have a public key created using three components, a large prime integer (pim), integer multiplicative group generator (gm), and the third part which is the generator raised to the power of s (gm^s) (with s being the private key). This third part is usually called the public key part/third part represented by puk = ge^s mod pim. These three constitute the public key.

In ElGamal cryptosystem, user Kofi generates the two keys, through the following steps;

i. He first generates/selects a random large prime integer (pim) which is usually 1024 bits.

ii. He then generates/selects randomly an integer multiplicative group generator (gm) which is in the range 1<gm<pim-1. This means for every co prime integer cp to pim, there should be an integer ks such that gm^{ks}=b mod cp.

iii. He goes ahead and then generates/selects a random integer s which is also in the range $1 \leq s \geq pim$. iv. The public key/third part is then calculated as puk = ge^s mod pim.

v. The ElGamal public key that user Kofi has created is displayed as (pim, ge, puk) and his private key to be used by him for decryption is s. The public key can then be sent to Ama using a private channel for communication between the two of them which may not be secure.

Now the ElGamal Encryption process that Ama will perform is as follows:

i. She has to first of all receive the public triplet key set (pim, ge, puk) and convert the message M as a set of numbers n1,n2,… whose range is between 1 and pim-1.

ii. In order for her to perform message M encryption, first of all a random integer number rn is generated and used to generate the ciphertext combination Ci1 and Ci2; iii. Thus for Ci1 and Ci2 computation;

$Ci1 = ge^{rn} \bmod pim$ (which is Ama's way of transmitting the random number rn to Kofi)

$Ci2 = (M^{*}puk^{rn}) \bmod pim$

iv. Ama then sends the ciphetexts (Ci1, Ci2) together as one ciphertext $Ci = n_{1,2,\ldots} * (ge^{s})^{rn} = n_{1,2,\ldots} * puk^{m}$ (thus for each of the n blocks of message to be sent) to Kofi.

Then for the decryption of ciphertext Ci comprising of (Ci1, Ci2) by Kofi with the use of the private key s, the following steps are taken;

i. He has to calculate an inverse modular of $(Ci1)^{rn}$ modulo pim, represented as $(Ci1)^{-rn}$, and usually called the decryption factor.

ii. The original message which Ama encrypted is gotten through the following calculation $M = Ci2 \times (Ci1)^{-rn} \bmod pim$, thus for every block of message $n_{1,2,\ldots} = (ge^{s})^{-rn} * Ci \bmod pim$.

Elgamal cryptosystem is relatively slow in speed when it comes to encryption of certain data like images and also gives a high overhead because of large ciphertext size (Singh and Kumar, 2012).

Another shortcoming of the Elgamal cryptosystem is message expansion. This is so because there is doubling of the transferred message. This shortcoming is from the choosing of a new random key for every block of the plaintext message (Andreas, 2005).

The key generation process of ElGamal is just a bit easier than that of RSA, nevertheless its processes of encryption and that of decryption tend to be very cumbersome than that of RSA (Singh and Kumar, 2012).

Moreover according to (Seurin and Treger, 2013) who produced a variant of Elgamal, they noticed that Elgamal cryptosystem is vulnerable when it comes to adjustive attacks on certain cipher texts of it, because it is manipulative. This means, assume with a certain cipher text (Ci1,Ci2) which translates to message M, someone (e.g. hacker) can effectively and easily create another corresponding ciphertext as *MKx* with its three components being (*Z, Kx, T*). Thus it cannot be secure according to IND-CCA2.

2.3 Elliptic Curve Cryptosystem Model

Koblitz Neil from Washington University and IBM's Miller Victor in 1985 created an elliptic curve theory based cryptography called the Elliptic Curve Cryptography (ECC) as a mechanism to alternatively implement public key cryptosystems. The elliptic curve equation in mathematics is used to do the encryption. ECC use smaller key sizes which can be efficient but may not be good for security yet not withstanding with its smaller key sizes, ECC offers some good security. Also with ECC, there is less power consumption yet may have the tendency of shortening the lifespan of batteries (Singh and Kumar, 2012). Even though the use of ECC started in 1985, yet from 2004 to 2005 Elliptic Curve Cryptography became widely used and one of its main uses is in resource constrained environments, like mobile networks and wireless ad-hoc networks.

A graph of a plane curve over a finite field (not real numbers) is the basic representation of an elliptic curve equation (Hakerson et al, 2004).

Figure 2.2 shows Elliptic Curve Representation

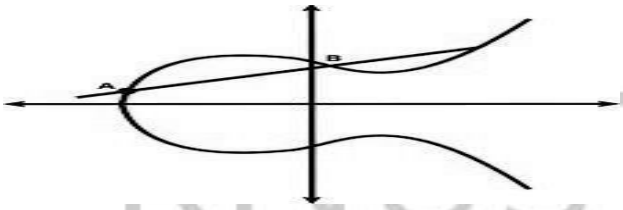


Figure 2.2 Elliptic Curve Representation

Elliptic curve has point values which satisfies the equation;

$v^2 = u^3+Au+B$; *thus the content point values being A and B.*

For ECC key generation, encryption and decryption of message M_{sg} to cipher text C_{ec1} and C_{ec2} the following steps with its subsequent computation is done;

i. Every message M_{sg} is first of all encoded as a point "C" on a suitable elliptic curve graph E_{cg} and another point "Z" is created on the curve also. This point generation involves a detailed implementation usually done by certicom.

ii. A Secret/Private Key S_k<S is selected and then used to calculate the public key P_k= S_kZ, with P_k being the Public key, S_k being the Private key and Z is the other point on the curve and S is a prime number.

iii. For encryption of message E_{cg} to get a cipher text C_{ec1} and C_{ec2}, the following computation is done ; first randomly select another random number r_2 <S, then compute $C_{ec1} = r_2$*Z and $C_{ec2} = C+r_2$* P_k. Thus C_{ec1} and C_{ec2} will be sent.

iv. For decryption of the two ciphertexts C_{ec1} and C_{ec2} back to Plaintext , the following computation is done; $E_{cg} = C_{ec2} - S_k$*C_{ec1}

A major strength of the ECC asymmetric cryptosystem is that its short key makes it faster in terms of encryption and decryption speed and subsequently makes its implementation and usage requires smaller computing power compared to its contemporaries. (Arya *et al*, 2015)

Neal Koblitz's ECC is a little based on the discrete logarithm problem. He worked on elliptic curves that are over multiplicative finite fields group. Thus he based the elliptic curve on the analog discrete logarithm theory/problem which is harder to compute as compared to the classical one. But one main setback of this approach/technique to the discrete logarithm problem is based on the complexity of the structure of the multiplicative group of a finite field of prime numbers and moreover how cumbersome its computation is (Koblitz, 1987).

Another disadvantage or limitation of ECC is an increased encryption text size and also complexity of the algorithm is high because it depends on complex equations (Singh and Bharti, 2013) and cost of implementation (Parmar *et al*, 2015).

In summary, with all these discussions and review of literature of some of the various asymmetric cryptosystems, it can be seen that all have some good strengths and some serious weakness. Moreover, according to (Lenstra *et al*, 2012), among the keys analyzed in their survey, of which both discrete logarithm and elliptic curve cryptosystems ones are approximately 45%, some duplicates were found with unauthorized owners and that this causes for concern/anxiety, because if these owners/users get to know, they can break each other's security. Also the (Lenstra *et al*, 2012) survey showed that approximately 55% of

the keys analyzed representing RSA keys also had about 1.1% of the prime numbers used to generate the keys not being truly random but also found duplicates when the Euclidean algorithm was used to factorize and reverse engineer the keys to get the prime numbers. Furthermore, according to some works citated in the thesis of (Yogita, 2016), some proposed that RSA can have ciphertext attacks, timing attacks, etc. Yet according to (Moore, 2012), a proposed flaw in RSA is not from the algorithm itself but from its implementation. Moreover, the thesis of (Farah et al,) concludes that RSA performs better in terms of performance than Elgamal and Paillier asymmetric cryptosystems although it slacked in some areas like high decryption time.

Also according to Kaminsky (2012) on a blog which he wrote in response to the "Ron is Wrong, Whit is Right" by (Lenstra *et al*, 2012), he stated that "single-secret" cryptosystems like the discrete logarithm cryptosystems (with even a recent example being the research by (Andrian *et al*, 2015) on Diffie-Hellman, in which flaws like TLS based attacks like logjam, different form level interceptions, etc were found) and elliptic curve cryptosystems and its variants have all failed in quite exceedingly larger numbers in the past compared to multi-secret key cryptosystems like RSA and its variants although RSA has also failed according to (Lenstra *et al*, 2012). Thus in this review of literature, the intention is not to debunk other literature but rather to really review the proposed problem/weakness and strengths of some of these asymmetric cryptosystems, based on theoretical or proposed assumptions and experimental evidence.

Therefore from this review of literature, it then shows that it has been proven that all the asymmetric cryptosystems in the three categories have some serious strengths but yet still

may or have been compromised in one way or another. Therefore it is eminent to develop a more efficient and secure asymmetric cryptosystem with good key management, then compare it through experimentation with existing ones to prove it is better than them and by so doing, also get to know empirically their strengths and weaknesses.

CHAPTER THREE

RESEARCH METHODOLOGY

3.0 Introduction

Asymmetric cryptosystems have many applications such as protecting data in transit and on storage. Major companies and individuals use most asymmetric cryptosystems to protect their data. This success demonstrates that, asymmetric cryptosystems provide much security, but not much detailed security and efficiency based experimental research has been conducted in the field of asymmetric cryptosystems especially an experimental research based on asymmetric cryptosystems' properties.

This brought the necessity for me to experimentally analyze and evaluate some common asymmetric cryptosystems, based on their properties like encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput,

throughput, randomness, key length, operation per instruction (O/I), by simulating those cryptosystems in this analysis in a test suite. The observed outcome will then be collected for further analysis.

Also a new and more efficient and secure cryptosystem is to be developed and strengths and weaknesses of common cryptosystems identified. This research's result will guide companies, corporations and individuals to make good decisions regarding the cryptosystem to choose and use to protect their data.

The asymmetric cryptosystems' algorithms were developed and implemented using JAVA IDE (JAVA Integrated Development Environment)

The research strategy I employed was experimental. An experiment consists of orderly carrying out a procedure having the intention to verify/establish or refute a hypotheses' validity. This implies that the research of experimentation aims at testing whether a hypothesis (whether new or existing) is based on facts (i.e. realistic or true) or vice versa, and moreover to show how particular processes or phenomenon really works. Therefore an experiment is to be conducted on the asymmetric cryptosystems, thus to comparatively test their properties in order to verify whether they meet the efficiency and security claims made about them.

The research study I used is a quantitative one. Quantitative Research is a research which places emphasis on quantities and measurements.

This research is about numeric data that is objective and deals with counting and measurements. This implies messages of a considerable number were encrypted and decrypted using the proposed cryptosystem and some popular asymmetric cryptosystems. I collected and analyzed the numeric results which are based on the empirical analysis of the properties of these cryptosystems.

3.1 Data Collection: Experiment and Observation

The research was conducted on a computer system by the use of a TEST SUITE that was developed using JAVA IDE (JAVA Integrated Development Environment)

In this research, I run asymmetric cryptosystems such as RSA, Elgamal and the proposed cryptosystem on the TEST SUITE using messages of different sizes, in order to check their main properties like; encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput, throughput, randomness, key length, operation per instruction (O/I). And in order to wholly test the data security level and efficiency of the various asymmetric cryptosystems, I also run different types and sizes of data (i.e. messages) many times and recorded the results.

In the collection of data in this research, the observation technique was also employed. Observation means taking a critical and careful look, study and note of something.

Therefore observation and experimentation techniques were the data collection techniques used in this study. Thus I conducted an experiment on the various properties of some common asymmetric cryptosystems with different messages on the proposed TEST SUITE and made full observation of the outcome afterwards and data was collected for analysis.

3.2 Framework for Data Analysis

I empirically analyzed collected data from the experiment by comparing the asymmetric cryptosystems based on the results obtained from tests run on their properties. Thus the data analysis of the results obtained from empirical experiment conducted on the various asymmetric cryptosystems using the test suite was done within a statistical framework.

3.3 The proposed asymmetric cryptosystem

The proposed cryptosystem is based on the model of RSA cryptosystem. In this cryptosystem 3 prime numbers were used instead of two prime numbers so that factorization becomes very difficult. By doing it this way, easy factorization problem will be eliminated and this will result in increasing security of the cryptosystem. This will make its factorization computation very complex for hackers.

Also, to increase the security of the cryptosystem, the public key (e,n) takes on four values instead of two, such that encryption exponent (e) takes on two values such that e = x/y and the modulus (n) also takes on two values such that, n= t/s.

The proposed cryptosystem is divided into three stages or processes which are;

i. Key Pair Generation ii.

Encryption Process

iii. Decryption Process.

Anyone who wishes to communicate using the encryption and decryption processes, have to first generate two keys. These key pair consists of the public keys and the private key.

The key pair generation process is described below;

First of all the proposed cryptosystem modulus (n) have to be generated as follows;

i. Choose 3 large primes, a, b,

c ii. Calculate n as n=a*b*c iii.

Represent n as t/s.

Then the derived number (e) which is the encryption exponent is calculated as follows;

i. The value of **e** should be greater than 1 and less than z where

[z = (a − 1) (b − 1) (c − 1)]

ii. There should be no common factor for the values e and z with the exception of 1.

That is, the gcd (greatest common divisor) of e and z is 1or e and z are co-prime.

(NB: This is done to help generate the private key easily by the use of e and z.) iii.

Represent “e” as “x/y” by mathematically computing for x and y.

The third part in the proposed cryptosystem key generation is the forming of the encryption or public key, which is done as follows;

i. The values (y, x, s, t) form the public key of the proposed cryptosystem. With this, two public keys are formulated as (y, t) and (s, x) and these are made public. ii. It will be difficult to find out the values of e and n as it is represented with the values (y, x) and (s, t) respectively, which are mathematically related. And even if the modulus n was known, the difficulty in factoring large numbers ensures the attacker will find it difficult in finite time to factor the product of the 3 primes (a, b, c) which were used to get the value n. This is a major strength of the proposed crptosystem in terms of data security.

The forming of the decryption/secret or private key is also done as follows;

i. The calculation of the secret/private key d is done from/using the public keys (y, t), (s, x) i.e. n and e. For a particular n and e, there exists a unique value d.

ii. The inverse of e modulo z gives the value of d. That is, d is a number less than z such that when is multiplied by e, its value is equal to 1 modulo z. Mathematically this is written as; $ed = 1 \bmod z$ iii. The number pair (d, n) forms the proposed cryptosystems private key and this is kept private.

Once these keys are gotten, the process of encryption and decryption are computationally straight forward and easy.

The plaintext is represented as series of numbers less than the modulus n hence the cryptosystem operates on a number modulo n.

The Proposed Encryption (on the plaintext/message (P)) in order to transform it to the cipher text (C)) is performed by the intended sender (Kofi) as follows;

i. Suppose a sender (Kofi) who has (y, t) and (s, x) as the public keys wishes to send information or text message to a recipient (Ama).

ii. The sender (Kofi) converts the plaintext P into a series of numbers that is less than n.

iii. In order to encrypt this plaintext P, the simple mathematical step below is used;

(Cipher text)$C = P^{x/y} \bmod (t/s)$ iv. This means that, the cipher text (C) is equal to the plaintext (P) multiplied by itself x/y times and then reduced to modulo n which is (t/s). This implies that (C) is also a number less than n.

Also The Proposed Decryption (on the cipher text (C) in order to transform it to the plaintext/message (P)) is performed by the intended recipient (Ama) of the message/plaintext (P) as follows;

i. The process of decryption is very straightforward. Suppose the receiver (Ama) of message/plaintext (P) has received its cipher text (C).

ii. The receiver (Ama) raises C to the power of his private key (d). The result modulo n then becomes the plaintext (P).

iii. Mathematically written as: Plaintext $(P) = C^{d} \bmod n$

Fig 3.1 is a block diagram of the proposed cryptosystems which summaries the mathematical computations involved in the proposed cryptosystem.

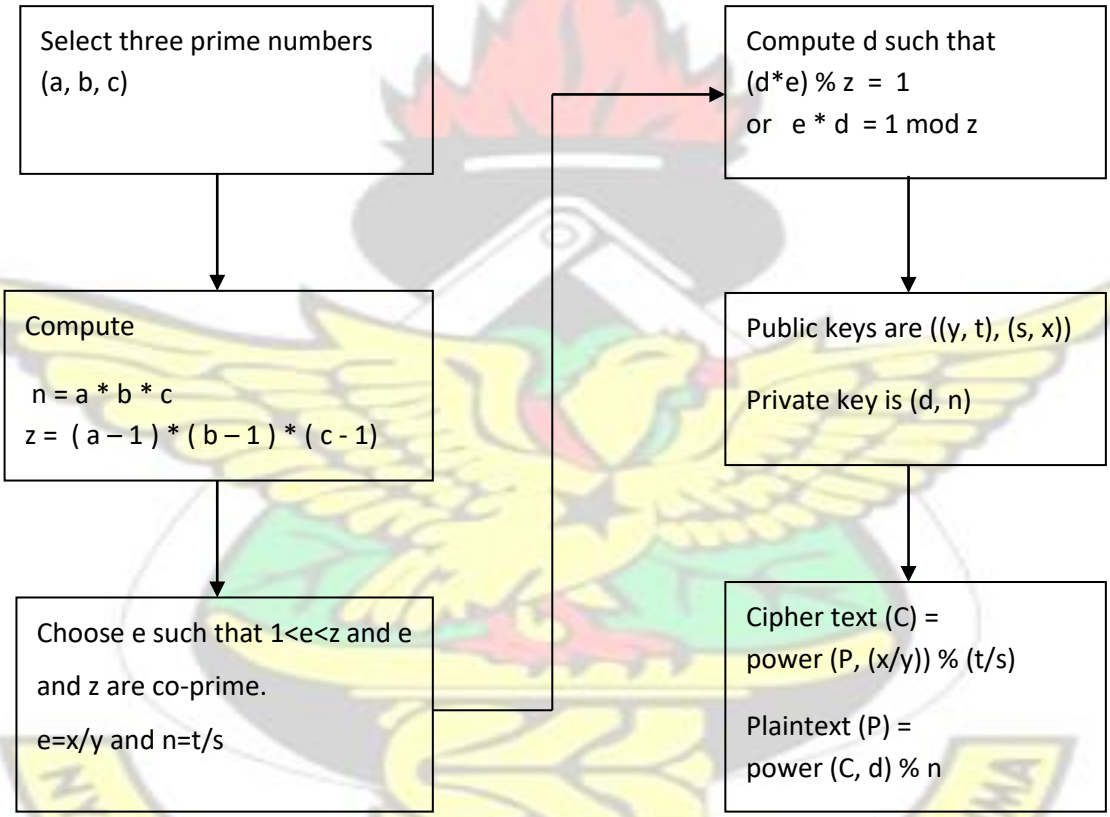


Figure 3.1 Block diagram of the proposed cryptosystem

Figure 3.2 is a flowchart of the proposed cryptosystem that shows implementation of the system.

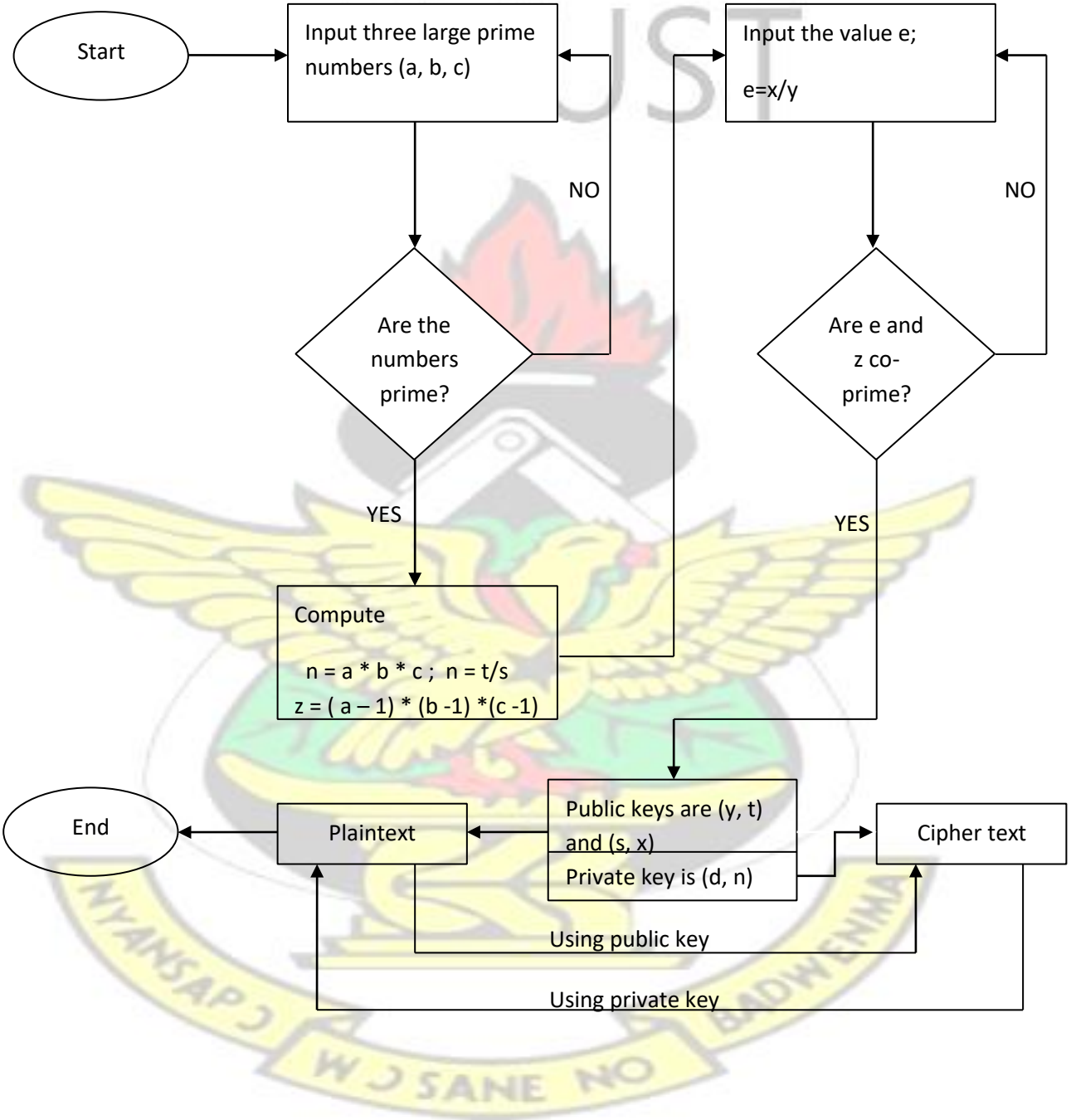


Figure 3.2 Flowchart of the proposed cryptosystem

3.4 The proposed test suite

The **TEST SUITE** I developed is a java based program and it is designed to test properties such as encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput, throughput, randomness, key length, operation per instruction (O/I) of RSA, Elgamal and the proposed cryptosystem.

3.4.1 Encryption Computation Time

This property tests the time an asymmetric cryptosystem takes to produce a cipher text from a plain text and is measured in nanoseconds. To test the encryption computation time for the various algorithms, I encrypted messages of different sizes using the various cryptosystems under study.

To get better results, I encrypted 50 messages, recorded the results and found the average encryption computation times for the various algorithms or the average time taken to encrypt the messages by the various cryptosystems. Moreover this same encryption computation time simulation test was performed on three groups of text with various text sizes (which are grouped according to their text sizes; thus small text sizes ranging from one byte to five hundred bytes(1B-500B), medium or average text sizes ranging from five hundred bytes to one kilobytes (500B-1KB) and then large text sizes ranging from one kilobytes upwards (1KB upwards)), to see how the cryptosystems under study would fare in terms of encryption computation time when they are used to encrypt various sizes of text. Therefore the cryptosystem that comes out with the smallest average encryption computation time is the best in terms of how fast the cryptosystem can encrypt data or

encryption speed and optimization. This will result in reduced encryption processing time (i.e. time for converting plaintext to cipher text) and less resource consumption when that cryptosystem is used, contributing to increased productivity or encryption work output.

3.4.2 Decryption Computation Time

This property tests the time taken by the various algorithms or cryptosystems to produce a plain text from a cipher text. This property is also measured in nanoseconds.

To test the decryption computation time for the various algorithms, I decrypted the 50 messages of different sizes that were encrypted earlier by the use of the various cryptosystems, recorded the results and found the average decryption computation times for each of the various algorithms or the average time taken to decrypt the messages by the various cryptosystems. Moreover this same decryption computation time simulation test was performed on three groups of text with various sizes (which are grouped according to their text sizes; thus small text sizes ranging from one byte to five hundred bytes(1B-500B), average text sizes ranging from five hundred bytes to one kilobytes (500B-1KB) and then large text sizes ranging from one kilobytes upwards (1KB upwards)), to see how the cryptosystems under study would fare in terms of decryption computation time when they are used to decrypt various sizes of text. The cryptosystem that comes out with the smallest average decryption computation time is the best in terms of decryption speed and is the most optimized cryptosystem.

Therefore when this cryptosystem is used, there will be reduced decryption processing time, less resource consumption, and in all leading to increased productivity or decryption work output.

3.4.3 Performance or speed

Performance property tests the time taken by the various algorithms or cryptosystems to perform encryption and decryption of the input data or text file. This constitutes a summation of both the average encryption computation time and average decryption computation time.

It shows the overall performance of the cryptosystem. This is important because the encryption computation time may be better in a cryptosystem but the decryption computation time may not or vice versa. This makes it important to measure this property to know the overall performance of the cryptosystem. What accounts for these differences between the encryption and decryption computation times is normally dependent on the type and structure of the asymmetric cryptosystem algorithm used, the size and differences of the keys used for both the encryption and decryption, how optimized the algorithm is and finally on the type of system (thus both hardware and (system and application) softwares) used in the implementation of the encryption algorithm.

To measure this property, I encrypted and decrypted 50 messages of different sizes by the use of the various cryptosystems under study, recorded the results and found the average performances for the various algorithms or cryptosystems. Also, this same performance simulation test was performed on three groups of text with various sizes(which are grouped according to their text sizes; thus small text sizes ranging from one byte to five hundred bytes(1B-500B), average text sizes ranging from five hundred bytes to one kilobytes (500B-1KB) and then large text sizes ranging from one kilobytes upwards (1KB upwards)), to see how the cryptosystems under study would fare in terms of performance (summation

of both encryption and decryption computation times) when they are used to encrypt and decrypt various sizes of text and cipher text respectively. The cryptosystem that comes out with the smallest average performance is the best in terms of how fast the cryptosystem can both encrypt and decrypt messages and how optimized the cryptosystem is.

This therefore implies when that cryptosystem is used there will be increased productivity because processing time is reduced and there is less resource consumption.

3.4.4 Encryption throughput

Encryption throughput property shows the total encrypted cipher text size as compared to the encryption time, thus a measure of the comparative encryption efficiency of a cryptosystem, which is computed as the amount of data that is encrypted in a given time. This is measured in byte per second (b/s).

To test the encryption throughput for the various cryptosystems, I encrypted messages of different sizes using the asymmetric cryptosystems under study and divided the various data sizes by the time taken to encrypt them.

To get accurate results, I encrypted 50 messages and divided the message or data sizes by their encryption computation times. I then recorded the results and found the average encryption throughputs for each of the various cryptosystems. Thus the bigger the encryption throughput value for a cryptosystem, the better the cryptosystem. This is because the bigger the encryption throughput, the more effective, efficient and more optimized the cryptosystem is and this will result in much work output (i.e. encryption) in a given time which contributes to increased productivity.

3.4.5 Decryption throughput

This property shows the total decrypted cipher text size as compared to the decryption time, thus a measure of the comparative decryption efficiency of a cryptosystem which is computed as the amount of data that is decrypted in a given time. This is measured in byte per second (b/s).

To test the decryption throughput for the asymmetric cryptosystems, I decrypted the 50 messages of different sizes that were encrypted earlier using the various cryptosystems and divided the various ciphertext sizes by the times taken to decrypt them. I then recorded the results and found the average decryption throughputs for the various cryptosystems. The bigger the decryption throughput value for a cryptosystem, the better that cryptosystem. This is because the bigger the decryption throughput, the more effective, efficient and more optimized the cryptosystem is and this will result in much work output (i.e. decryption) in a given time which contributes to increased productivity.

3.4.6 Throughput

This property represents the overall throughput of a cryptosystem. It shows the overall efficiency and how productive a cryptosystem is. It comprises the summation of the encryption and decryption throughput values of a cryptosystem.

To get accurate results, I run 50 messages, found the overall average throughput and recorded for further analysis. The bigger the overall average throughput value for a cryptosystem, the better the cryptosystem. This is because the bigger the overall throughput, the more efficient and more optimized the cryptosystem is and this will result in much work output (i.e. encryption and decryption) or increased productivity.

3.4.7 Randomness

This property shows how random each encrypted data/ciphertext sequence becomes each time its plaintext is encrypted. Asymmetric cryptosystems should be random algorithms (i.e. random algorithm or function generates or produces random outputs each time it is fed/inputted with a particular given input); in the sense that any message or plain text that is passed through a cryptosystem must produce different cipher text no matter how many times the same message is fed into the cryptosystem. For instance a particular message encrypted 50 times must produce or generate 50 different outputs or cipher texts. The strength of a cryptosystem is proportional to the degree of randomness of the encrypted data. This is because it makes it difficult for hackers to derive meaning of another cipher text produced from the same plaintext whose previous cipher text was known by them.

To measure this property, I encrypted a particular message fifty (50) times using the selected asymmetric cryptosystems under study. I recorded and observed the cipher texts whether they are all different for the same plaintext which was encrypted. I checked this phenomenon for each of the cryptosystems under study whether they all exhibited that "randomness" behaviour. For a particular cryptosystem to produce different cipher texts when encrypted a lot of times, then it means that it is not deterministic and this help increase security of the cryptosystem.

KNUST

3.4.8 Key length

This property deals with the size/length of the public/encryption key. This measures the lower bound strength of cryptosystems. The general rule for asymmetric cryptosystems is that the longer the public key size, the better the cryptosystem in terms of security.

To measure this property for the various cryptosystems, I recorded about 50 key sizes of public keys that were used to encrypt different messages and the average key sizes of the various asymmetric cryptosystems were taken for further analysis. The strength of an encryption algorithm is directly proportional to the key size. Hence an increased or higher key size results in increase in the security of the cryptosystem, in the sense that, hackers who gets access to the key and tries to use it to find the plaintext when they have the cipher text will find it very difficult/cumbersome (e.g. in performing computations)when the key size is high.

Multiple keys: The more the number of public keys (i.e. used for one encryption) and the number of values which are related mathematically in the multiple public keys the more cumbersome/difficult and time consuming it will be for an attacker to figure out how to compute the private key from the multiple public keys. This therefore contributes to the security or strength of the cryptosystem.

Cipher text length: The longer the cipher text, the more it is considered to be secured because the text will be plenty and the attacker will have to do much work and consume much time to figure out the plaintext.

3.4.9 Operation per instruction (O/I)

This property shows how secure an encrypted data is by using a particular cryptosystem to encrypt that data. It measures the upper bound strength of cryptosystems. This is an estimate of the amount of work that is required by hackers to defeat a cryptosystem and it shows the attack resistance level of a cryptosystem.

The number field sieve was used to estimate the number of simple arithmetic operations that is needed to factor an integer say n, thus it predicts the difficulty to factor large numbers (i.e. modulus of the various cryptosystems). It is considered in this paper that the difficulty in performing discrete logarithm is proportional to the difficulty in factoring large numbers. This document sticks with one operation per one instruction (O/I),

(Orman and Hoffman, 2004) which uses the formula stated below;

O/I= 6 * 10^(-16) * e^(1.92 * cubrt(ln(n) * (ln(ln(n)))^2)) (Lenstra and Lenstra, 1993), where **n** is the modulus of an asymmetric cryptosystem, and **ln** is the natural logarithm and **e** is the base of the natural logarithm.

To test this property, I encrypted messages of different sizes using the various cryptosystems, thus I encrypted 50 messages, recorded the results (i.e. the modulus of the cryptosystems) and used it to find the average O/I for the various cryptosystems. The cryptosystem that comes out with a higher value in terms of O/I will be the one to be able to provide higher data security (Orman and Hoffman, 2004). A higher value of O/I means

a hacker will have to do a lot of work to be able to decrypt a message (one instruction) that is encrypted by a particular cryptosystem or break that particular cryptosystem.

CHAPTER FOUR

DISCUSSIONS

4.0 Introduction

In chapter four, data/information that I gathered from the proposed test suite is analyzed. The findings gathered (from the simulation done using the test suite) and their explanations, with comparative analysis of the proposed asymmetric cryptosystem against the various other existing asymmetric cryptosystems are made.

The asymmetric cryptosystems that I analyzed are;

i. RSA

ii. Elgamal iii. The proposed cryptosystem.

Properties of the asymmetric cryptosystems, whose data I used in analyzing the cryptosystems under study are;

i. Encryption computation

time ii. Decryption computation

time iii. Performance iv.

Encryption throughput

v. Decryption throughput vi.

Throughput vii. Randomness

viii. Key length ix. Operation per

instruction (O/I).

4.1 Encryption Computation Time Comparisons

Considering figure 4.1a, for the fifty simulations I performed, on the average, it took 5963215.84 nanoseconds to encrypt messages using Elgamal cryptosystem making it the one with the highest average encryption computation time among the three cryptosystems used in the simulation test. RSA cryptosystem follows with an average encryption computation time of 3960663.58 nanoseconds which is lesser than Elgamal's average decryption computation time but higher than the proposed cryptosystem's average encryption computation time which is 2369631.76 nanoseconds. The proposed cryptosystem has the least encryption computation time as compared to Elgamal and RSA cryptosystems' encryption computation times. This clearly shows or means that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in the encryption of messages compared to the other two cryptosystems, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time) compared to Elgamal and RSA cryptosystems.

Moreover considering figure 4.1b, the three cryptosystems under study were used to encrypt three groups of text (which are grouped according to their text sizes);and from that simulation/analysis test, it can be seen that Elgamal cryptosystem still came out with the highest/largest average encryption computation times for all the three groups of different sizes of text encrypted, thus1765784.4 nanoseconds for small text sizes, 4784357.9 nanoseconds for medium text sizes and 231000580.9 nanoseconds for large text sizes. With

RSA once again following with a value of 691641.7nanoseconds for small text sizes, 3620585.6 nanoseconds for medium text sizes and 164377105.7 nanoseconds for large text sizes and the proposed cryptosystem still having the least average encryption computation times for all the three groups of text sizes thus 307847.3 nanoseconds for small text sizes, 1531408.6 nanoseconds for medium text sizes and 103418651.7 nanoseconds for large text sizes. This clearly shows that the proposed cryptosystem encrypts any size of text faster than the other cryptosystems.

Therefore, if the priority is to encrypt messages of any size faster, then the best cryptosystem to choose is the Proposed Cryptosystem. This is because it has better encryption computation time and uses less computing resources and therefore it stands from the simulation analysis as the one that uses the least computing resources and also the fastest in terms of encrypting text followed by RSA cryptosystem and then Elgamal cryptosystem.

KNUST

Figure 4.1a is a chart showing the three asymmetric cryptosystems under study and their average encryption computation times.

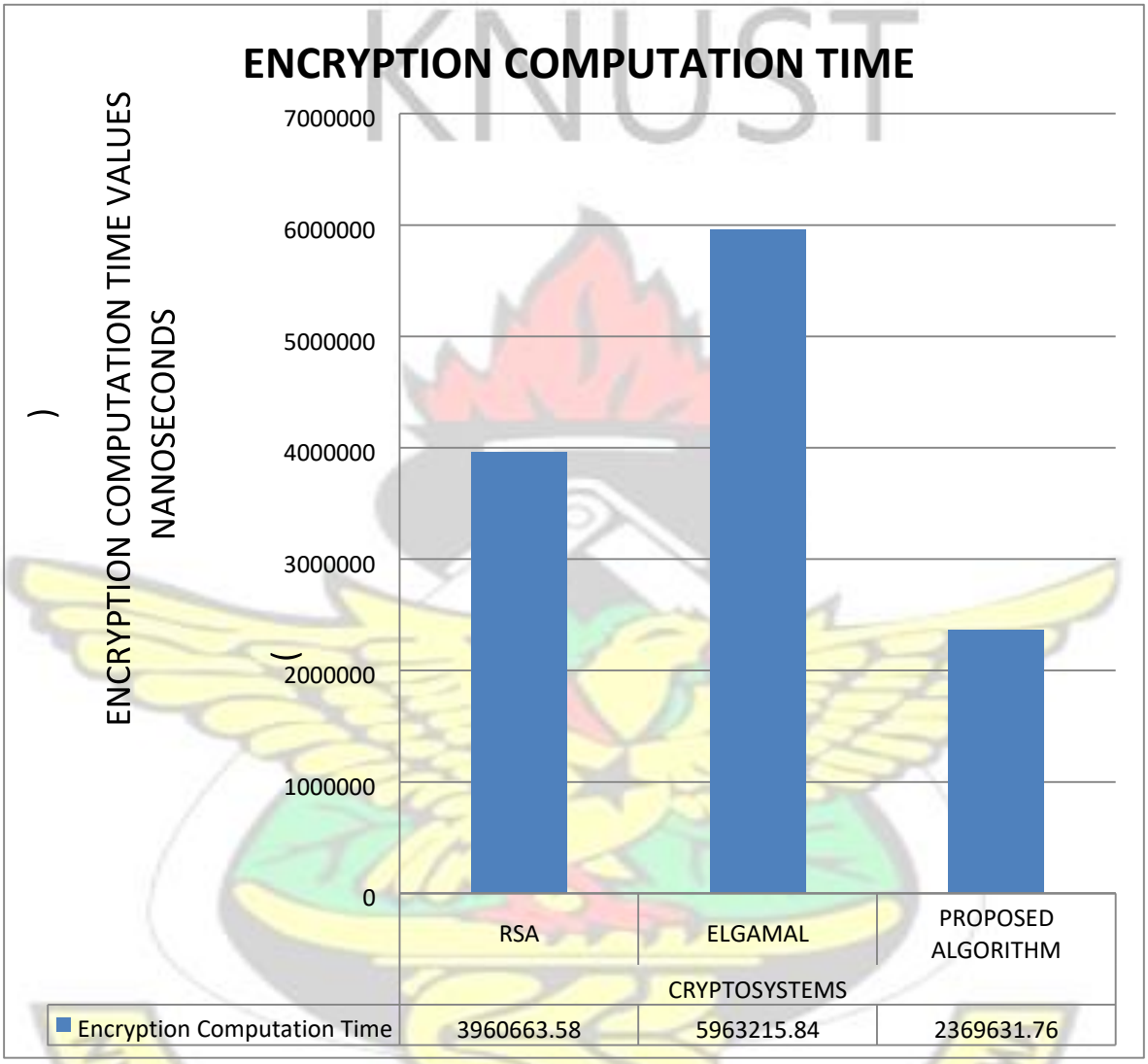
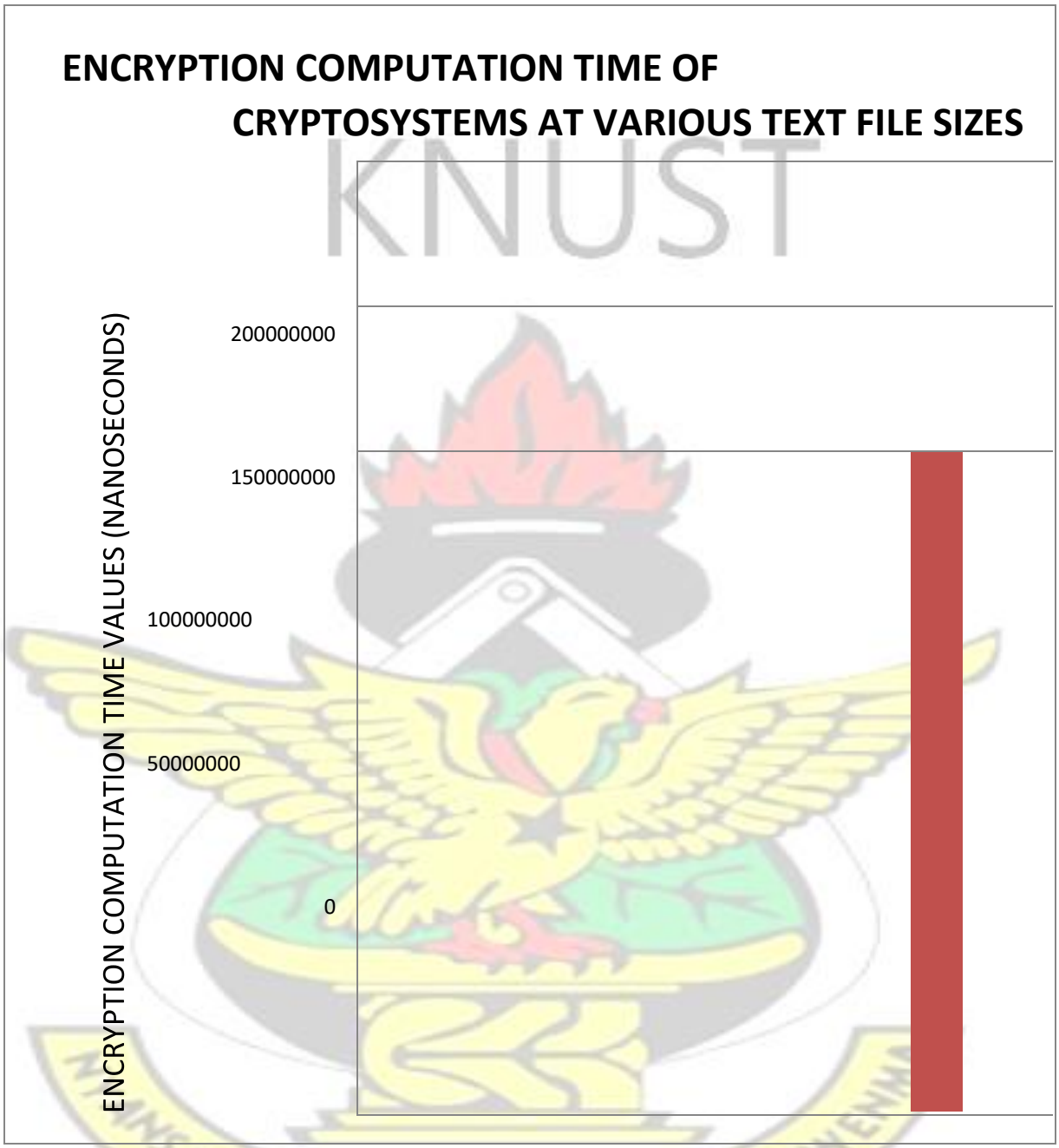
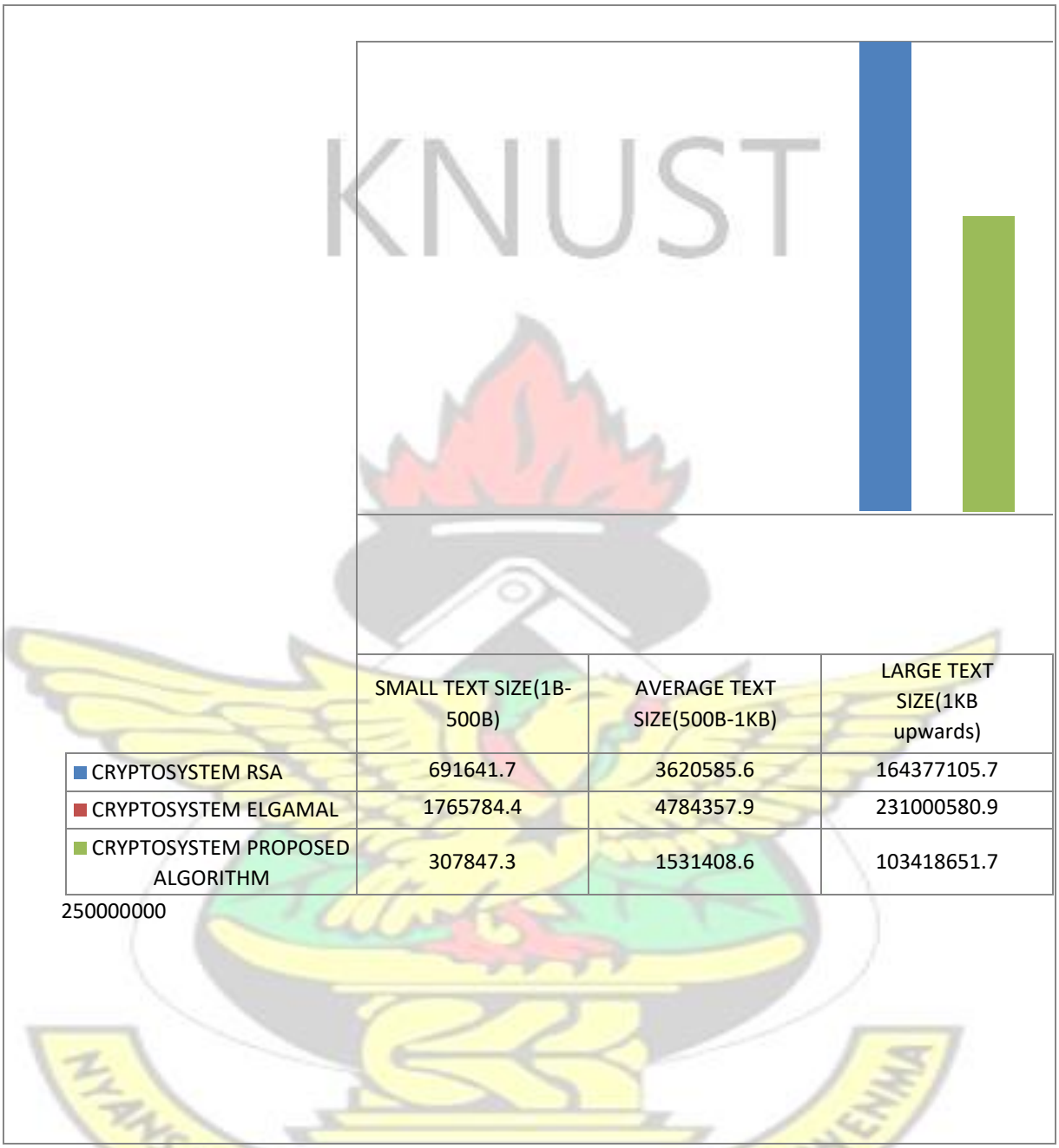


Figure 4.1a Cryptosystems and their average encryption computation time in nanoseconds

Figure 4.1b is a chart showing the three asymmetric cryptosystems under study and their average encryption computation time in nanoseconds for the three groups of text sizes.


ENCRYPTION COMPUTATION TIME OF CRYPTOSYSTEMS AT VARIOUS TEXT FILE SIZES
ENCRYPTION COMPUTATION TIME VALUES (NANOSECONDS)
200000000
150000000
100000000
50000000
0

KNUST



	SMALL TEXT SIZE(1B-500B)	AVERAGE TEXT SIZE(500B-1KB)	LARGE TEXT SIZE(1KB upwards)
CRYPTOSYSTEM RSA	691641.7	3620585.6	164377105.7
CRYPTOSYSTEM ELGAMAL	1765784.4	4784357.9	231000580.9
CRYPTOSYSTEM PROPOSED ALGORITHM	307847.3	1531408.6	103418651.7

Figure 4.1b Cryptosystems and their average encryption computation time in nanoseconds for various text sizes.

KNUST

4.2 Decryption Computation Time Comparisons

Taking figure 4.2a into consideration, for the fifty simulations I performed, on the average, it took 9936733.4 nanoseconds to decrypt ciphertext (of already encrypted messages) using Elgamal cryptosystem making it the one with the highest average decryption computation time among the three cryptosystems used in this simulation test/experiment. RSA cryptosystem follows with an average decryption computation time of 2098261.82 nanoseconds which is lesser than Elgamal's average decryption computation time but higher than the proposed cryptosystem's average decryption computation time which is 1442936.32 nanoseconds, the least average decryption computation time compared to Elgamal and RSA cryptosystems' average decryption computation time. This clearly shows or means that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in the decrypting/converting of ciphertext to plaintext compared to RSA and the proposed cryptosystems, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time) in the decryption of ciphertexts compared to Elgamal and RSA cryptosystems.

Moreover considering figure 4.2b, when the three cryptosystems under study were used to decrypt three groups of cipher text (which are grouped according to the text sizes), it was realized from the simulation analysis, that Elgamal cryptosystem still came out with the highest/largest average decryption computation times for all the three groups of different sizes of ciphertexts decrypted with a value of 3105562.4 nanoseconds for small text sizes, 7670060.7 nanoseconds for medium text sizes and 573076601 nanoseconds for large text

sizes. And RSA once again following with a value of 152518 nanoseconds for small text sizes, 1999398.6 nanoseconds for medium text sizes and 123330012.4 nanoseconds for large text sizes and the proposed cryptosystem once again came out with the least average decryption computation times for all the three groups of ciphertexts decrypted with a value of 96963.6 nanoseconds for small text sizes, 1165448.9 nanoseconds for medium text sizes and 103983884 nanoseconds for large text sizes. This clearly shows that the proposed cryptosystem decrypts any size of text faster than RSA and Elgamal cryptosystems.

Therefore, if the priority is to decrypt ciphertext/messages of any size faster, then the best cryptosystem to choose is the Proposed Cryptosystem. This is because it has better decryption computation time and uses less computing resources, thereby contributing to higher/increased productivity or decryption work output. It therefore stands out from the simulation analysis as the one that uses the least computing resources and also the fastest in terms of decrypting message/ciphertext followed by RSA cryptosystem and then Elgamal cryptosystem.

KNUST

Figure 4.2a is a chart showing the three asymmetric cryptosystems under study and their average decryption computation times.

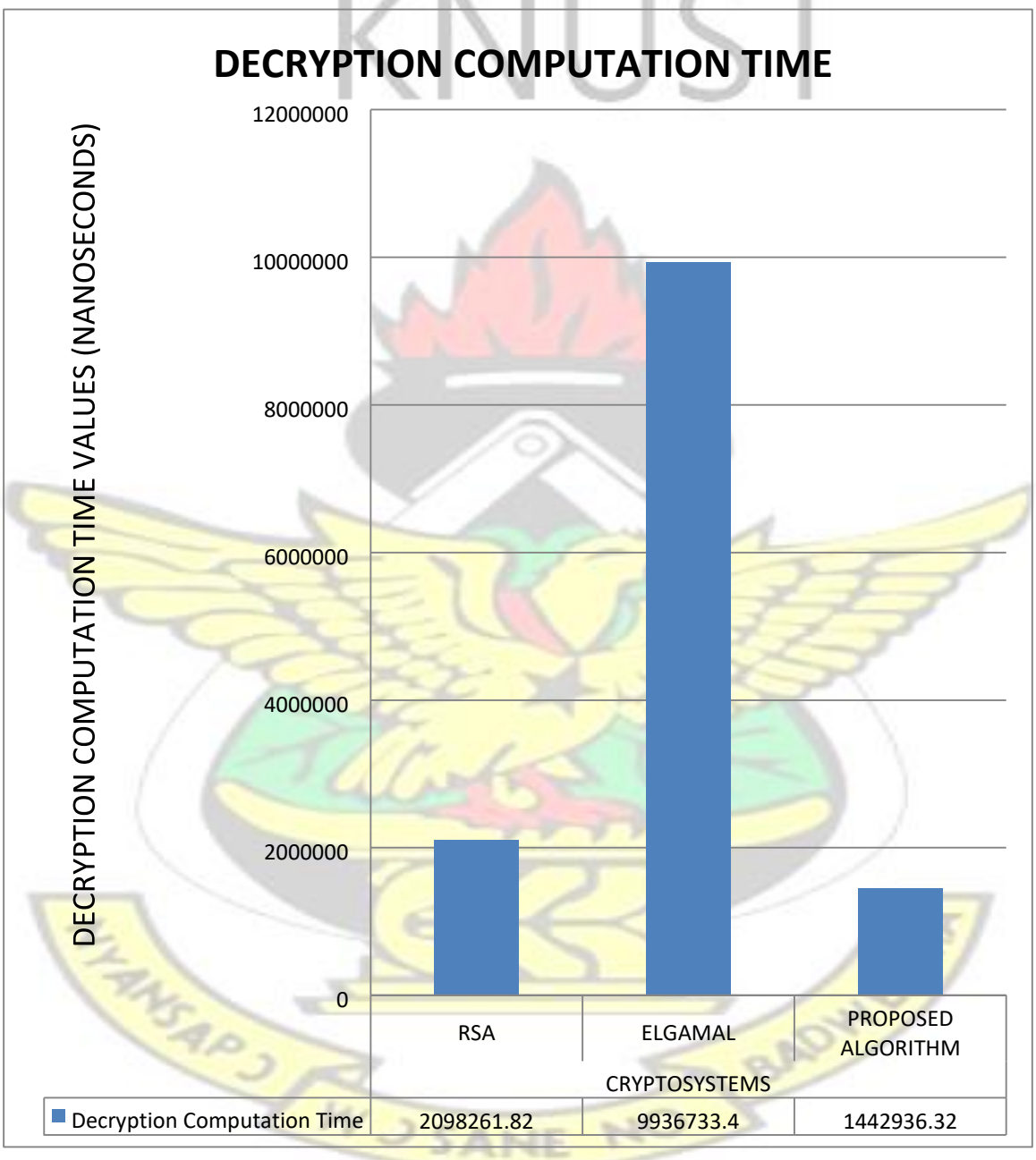
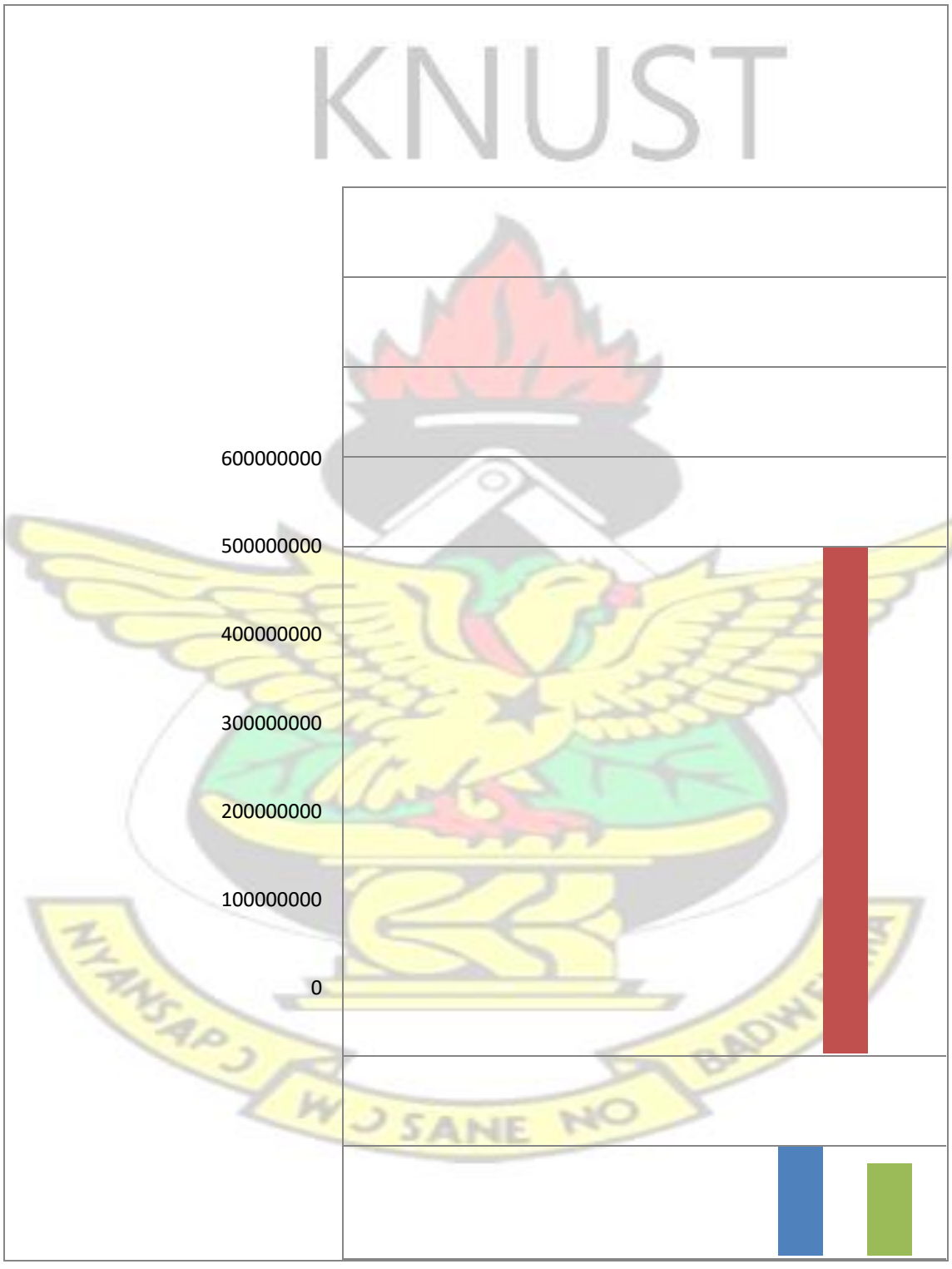


Figure 4.2a Cryptosystems and their average decryption computation time in nanoseconds.

Figure 4.2b is a chart showing the three asymmetric cryptosystems under study and their average decryption computation times in nanoseconds for the three groups of text sizes.



KNUST

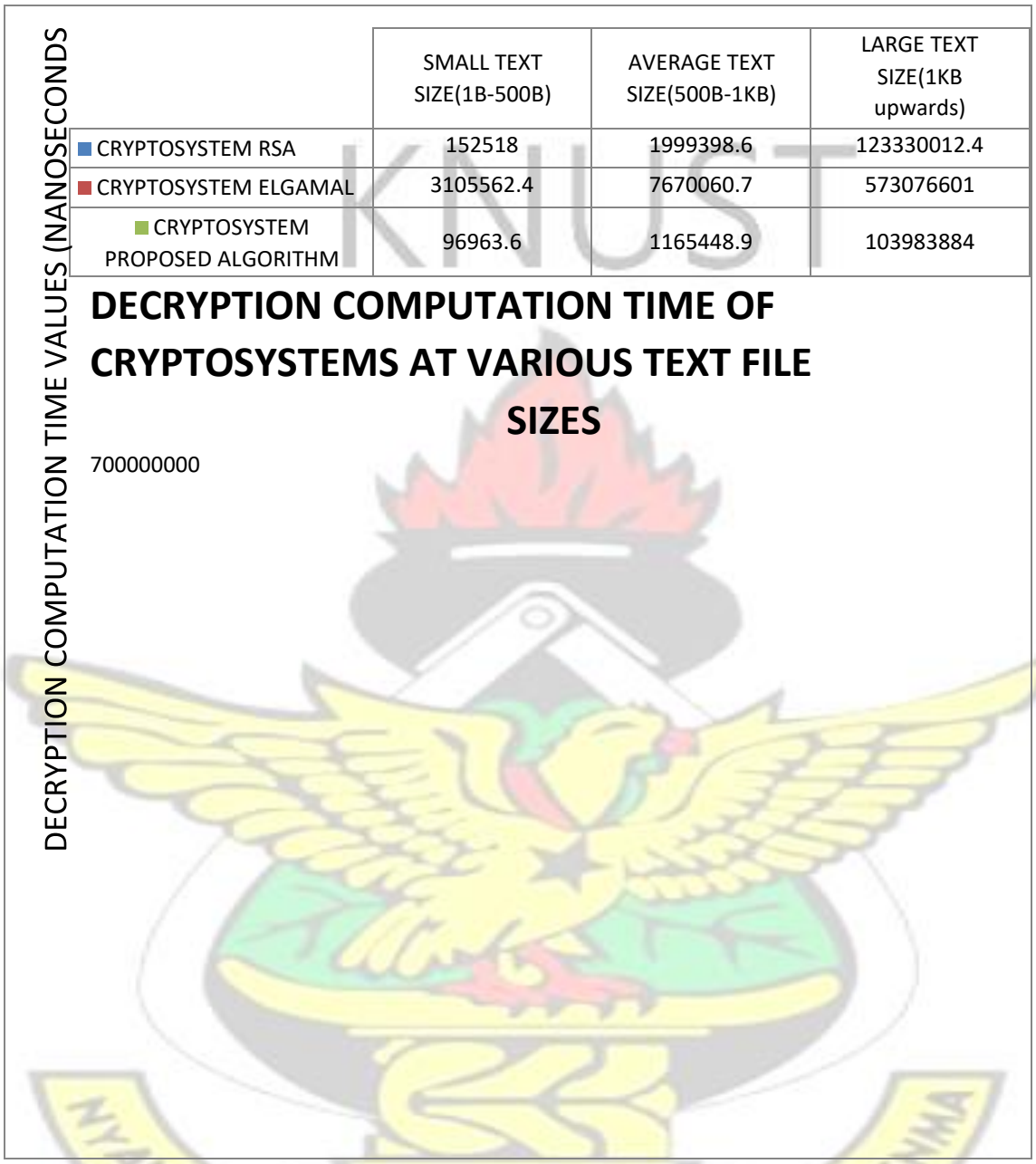


Figure 4.2b Cryptosystems and their average decryption computation time in nanoseconds for various text sizes.

4.3 Performance or Encryption and Decryption Speed Comparisons

Considering figure 4.3a, for the fifty simulations performed, the total average performance

(i.e. addition of the average encryption and average decryption computation times) of Elgamal cryptosystem is 16101573.24 nanoseconds, meaning an Elgamal cryptosystem takes more time to do both encryption of a message and its subsequent ciphertext decryption, thus making it the one with the highest/largest performance value among the three cryptosystems used in the experiment/test. RSA cryptosystem follows Elgamal with an average performance computation time of 6058925.4 nanoseconds which is lesser than Elgamal's performance value, but higher than the proposed cryptosystem performance value (i.e. addition of encryption and decryption computation times) which is 3812567.6 nanoseconds, the least performance value compared to Elgamal and RSA cryptosystems' performance values. This simulation of performance comparisons clearly shows that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in the encryption and decryption of messages compared to RSA and the proposed cryptosystem, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time) in the encryption of messages and decryption of ciphertext compared to Elgamal and RSA cryptosystems.

Moreover considering figure 4.3b, comparing the performance values (i.e. addition of both encryption and decryption computation times) of the three cryptosystems under study when they were used to encrypt and decrypt three groups of text (which are grouped according to their text sizes (i.e. small, medium/average and large)), Elgamal cryptosystem still came out with the highest/largest performance values for all the three groups of different sizes of text with values of 2435673.4 nanoseconds for small text sizes, 6227209.3 nanoseconds for

medium text sizes and 402038591 nanoseconds for large text sizes. RSA once again followed with a value of 422079.85 nanoseconds for small text sizes, 2809992.1 nanoseconds for medium text sizes and 143853559.1 nanoseconds for large text sizes and the proposed cryptosystem still had the least performance value for all the three groups of text sizes, thus 202405.45 nanoseconds for small text sizes, 1348428.75 nanoseconds for medium text sizes and 103701267.9 nanoseconds for large text sizes.

This performance property is essential because there can be differences in both the encryption and decryption computation times which in the case of the test I conducted was fundamentally and firstly as a result of the system (thus both hardware and system softwares) used in the implementation of the cryptosystems under study. Secondly from the simulation it was realized that what accounted for these differences in encryption and decryption computation times was also due to unoptimized code in the algorithms of the Elgamal and RSA cryptosystem used. Moreover differences in the sizes and differences in randomness of their public and private keys were the other two factors. But for the proposed cryptosystem the differences in encryption and decryption computation times are as a result of how the encryption side of the algorithm and multiple keys work.

Therefore, if the priority is for faster encryption and decryption of any size of text the proposed cryptosystem stands as the best choice or option to choose because from the simulation test analysis it is the fastest in both encryption and decryption and uses less computing resources followed by RSA cryptosystem and then Elgamal cryptosystem.

Figure 4.3a is a chart of performance comparison of the 3 asymmetric cryptosystems.

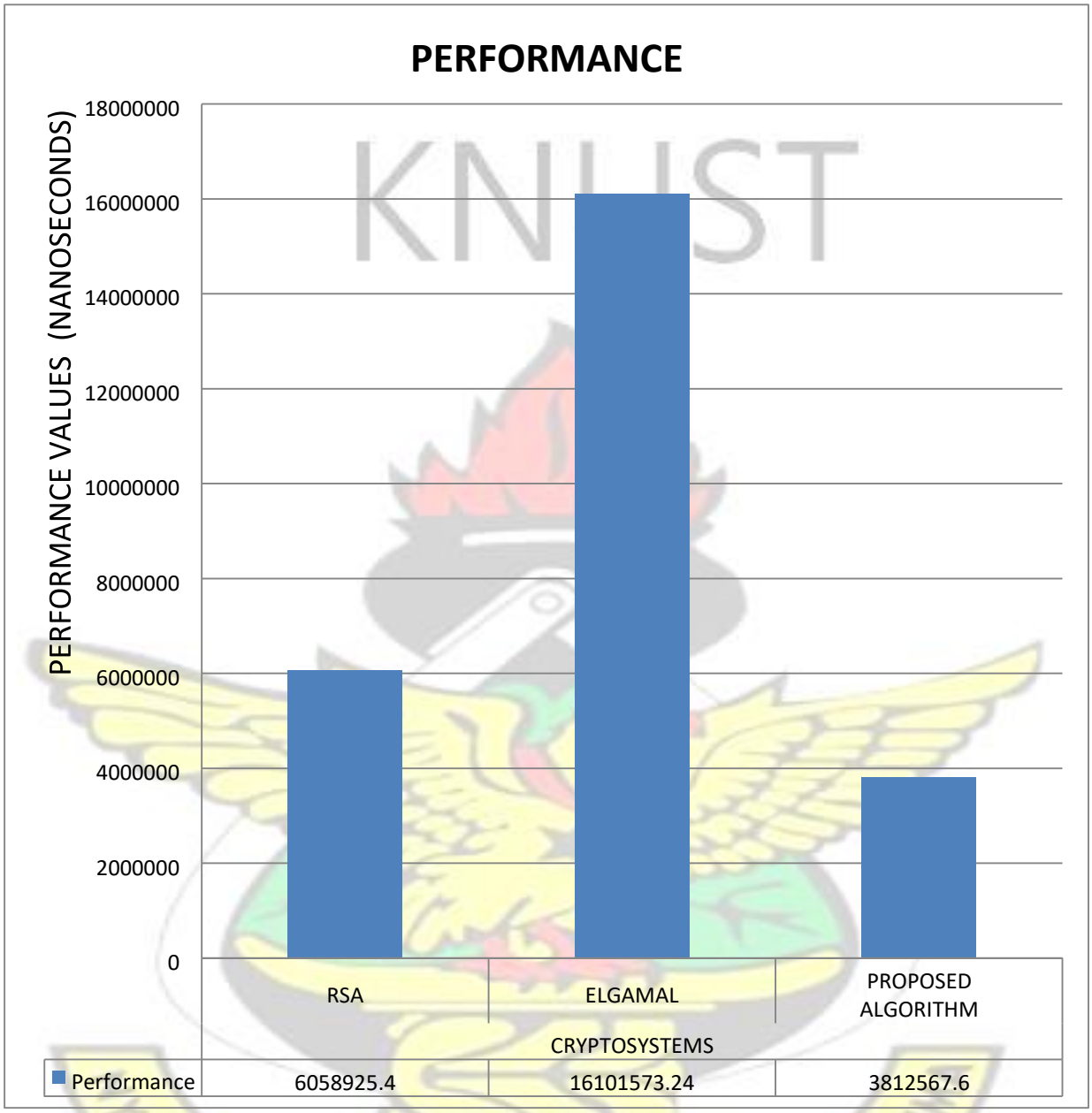


Figure 4.3a Cryptosystems and their average performance in nanoseconds

Figure 4.3b is a chart showing the three asymmetric cryptosystems under study and their average performances in nanoseconds for the three groups of text sizes.

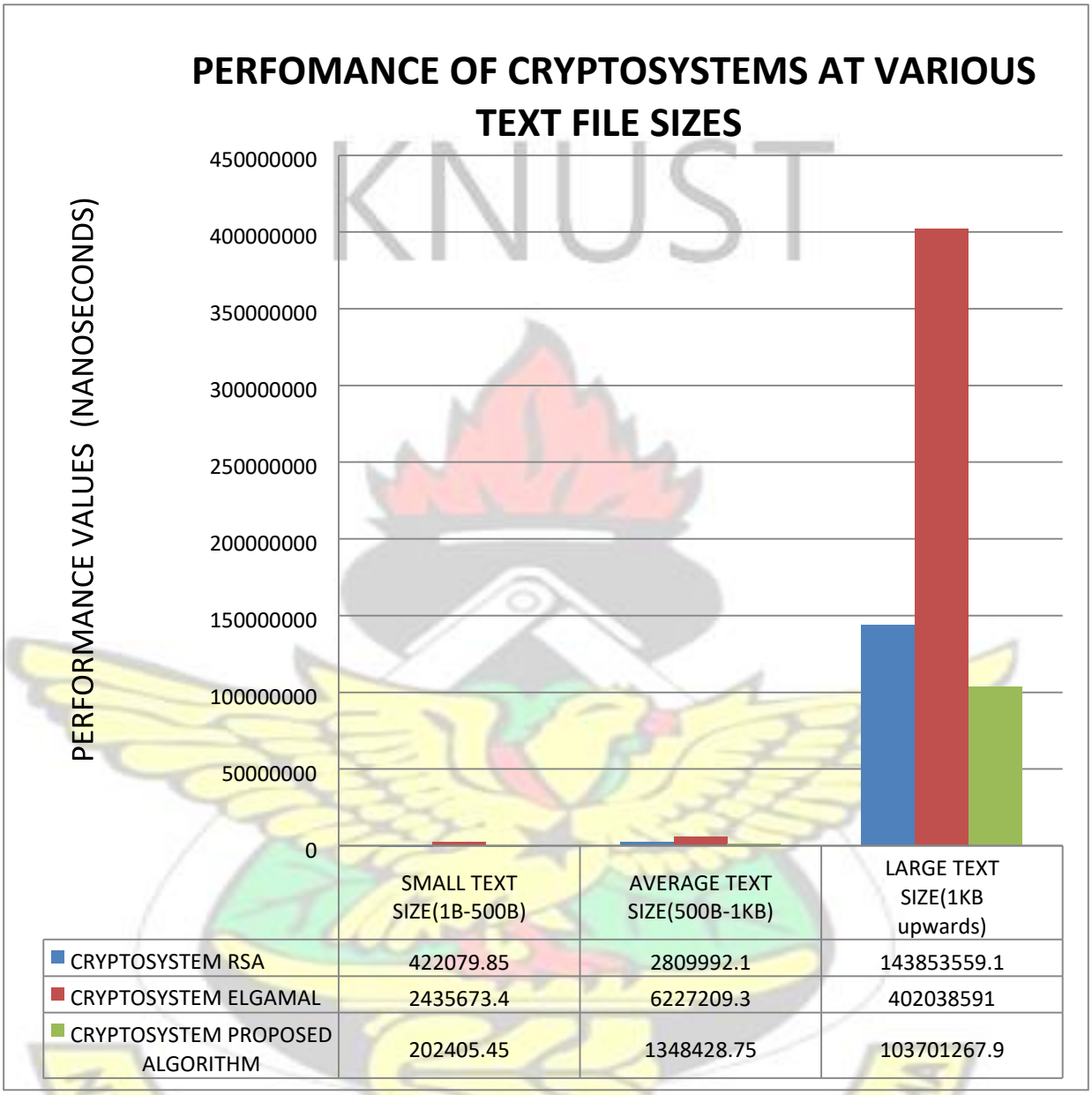


Figure 4.3b Cryptosystems and their average performance in nanoseconds for various text sizes

4.4 Encryption throughput Comparisons

Looking at figure 4.4, for the fifty encryption throughput (i.e. the amount of data/messages that can be encrypted within a specified time) simulations I performed, on the average, Elgamal cryptosystem recorded 18897.58 b/s (bytes per second) during the encryption of messages making it the one with the lowest average encryption throughput among the three cryptosystems used in the simulation test. RSA cryptosystem follows with an average encryption throughput of 996992.3 b/s (bytes per second) which is higher than Elgamal's average encryption throughput but lesser than the proposed cryptosystem's average encryption throughput which is 2241522 b/s (bytes per second), thus the highest encryption throughput compared to Elgamal and RSA cryptosystems' encryption throughputs. This clearly shows or means that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in the encryption of messages of any size and therefore less efficient (i.e. can do less work/encrypt less data in a given amount of time), compared to the other two cryptosystems, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time) in encrypting messages of any size and is therefore more efficient (i.e. can do more work/encrypt more data in a given amount of time) compared to Elgamal and RSA cryptosystems.

Therefore, if the priority is to encrypt messages of any size faster and efficiently thus to give higher encryption work output, then the best cryptosystem to choose is the Proposed Cryptosystem.

Figure 4.4 depicts the average encryption throughput comparison of the three asymmetric cryptosystems under study.

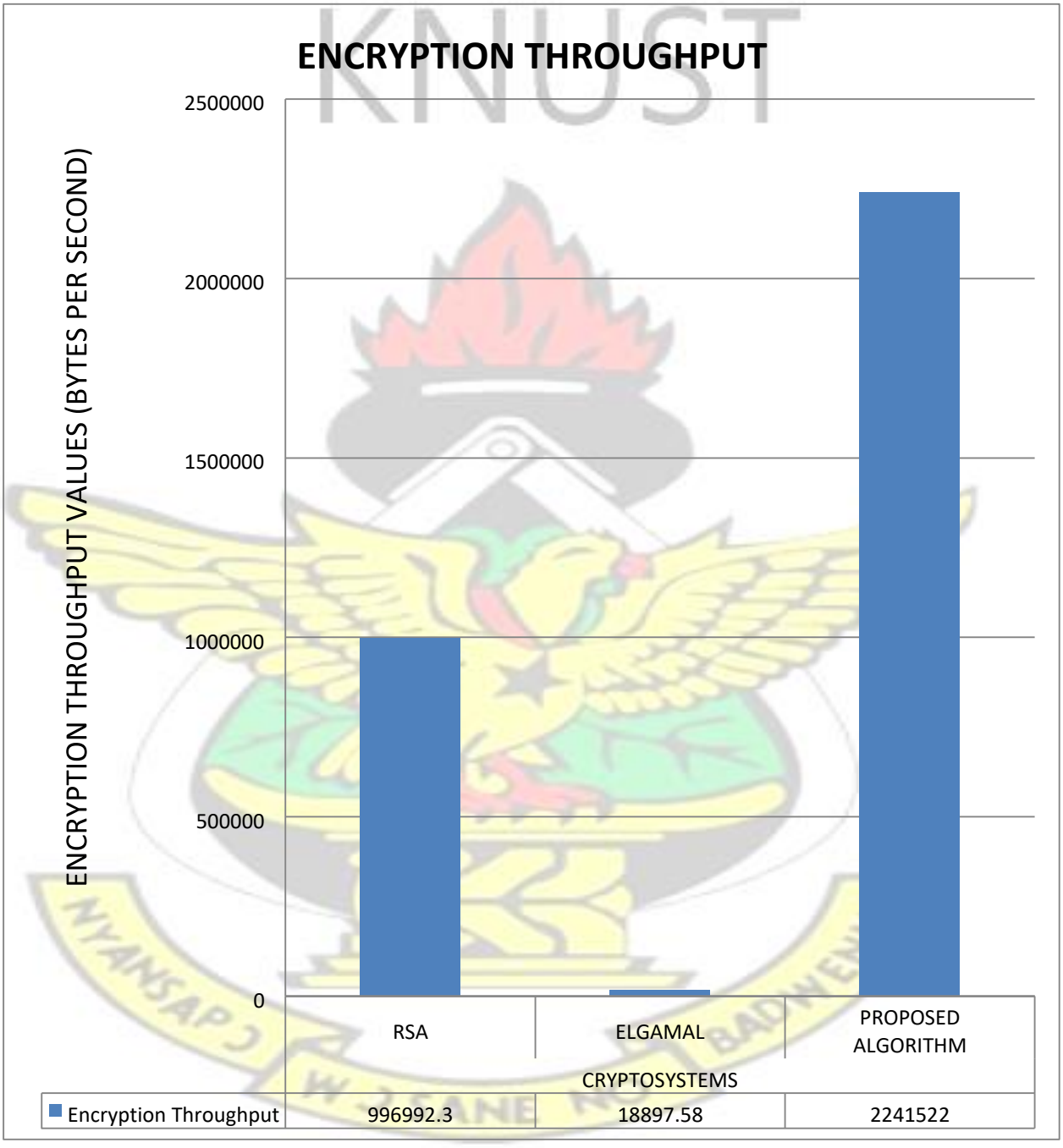


Figure 4.4 Cryptosystems and their average encryption throughputs in bytes/second

4.5 Decryption throughput Comparisons

Also considering figure 4.5, for the fifty decryption throughput (i.e. the amount of data/ciphertext that can be decrypted within a specified time) simulations I performed, on the average, Elgamal cryptosystem recorded 15917.8b/s (bytes per second) during the decryption of messages making it the one with the lowest average decryption throughput among the three cryptosystems used in the simulation test. RSA cryptosystem follows with an average decryption throughput of 78330.06b/s (bytes per second) which is higher than

Elgamal's own but lesser than the proposed cryptosystem's average decryption throughput which is 84029.26 b/s (bytes per second), thus making it the highest average decryption throughput compared to Elgamal and RSA cryptosystems' decryption throughputs. This clearly shows or means that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in decrypting ciphertexts/messages of any size and therefore less efficient (i.e. can do less work/decrypt less data in a given amount of time), compared to the other two cryptosystems, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time)) in decrypting ciphertexts/messages of any size and is therefore more efficient (i.e. can do more work/decrypt more data in a given amount of time) compared to Elgamal and RSA cryptosystems.

Therefore, if the priority is to decrypt ciphertexts of any size faster and efficiently thus to give higher decryption work output, then the best cryptosystem to choose is the Proposed Cryptosystem.

Figure 4.5 depicts the average decryption throughput comparison of the three asymmetric cryptosystems under study.

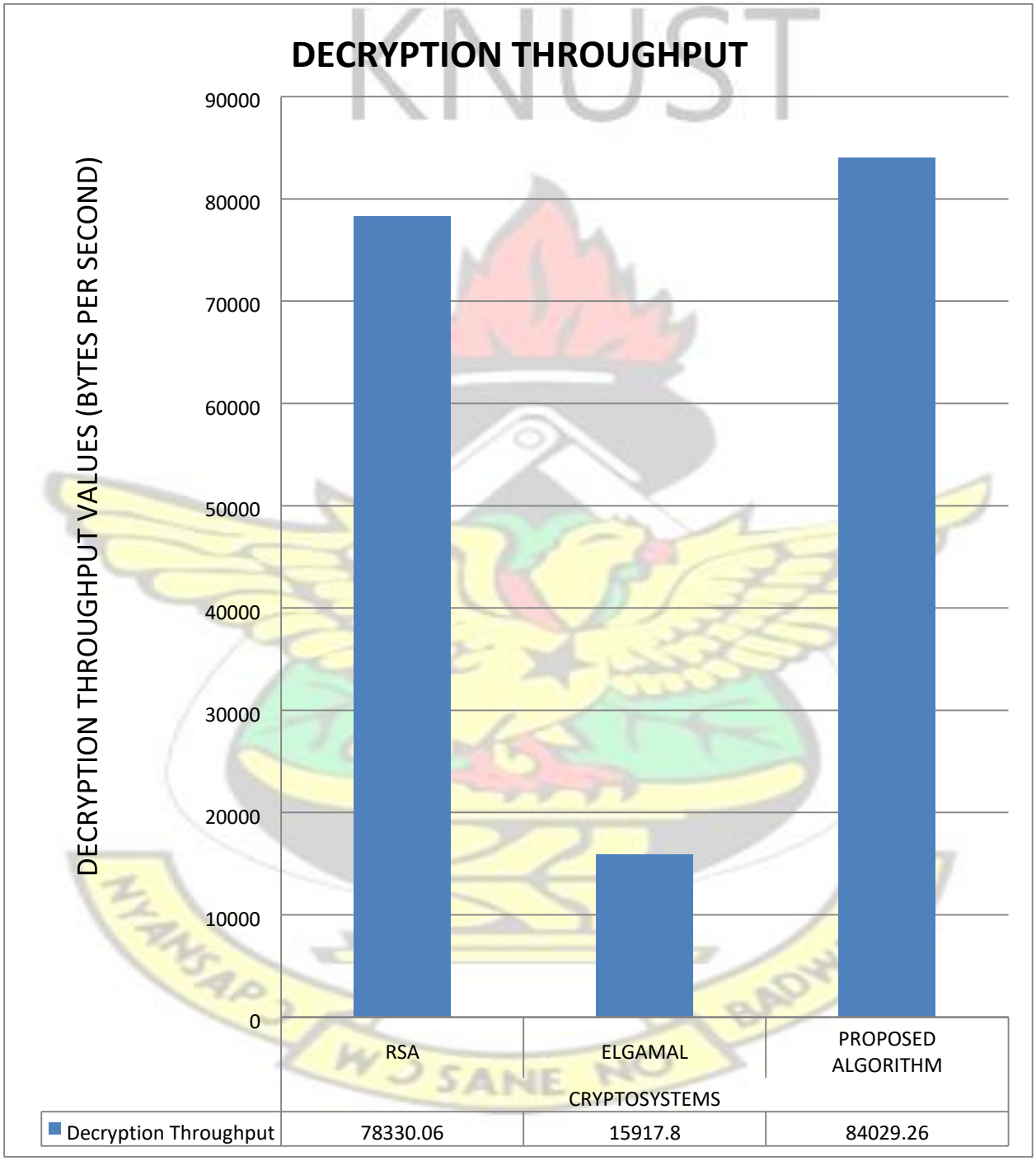


Figure 4.5 Cryptosystems and their average decryption throughput in bytes/second

4.6 Throughput Comparisons

Also considering figure 4.6, for the fifty total throughput (i.e. the addition of both the amount of data/messages that can be encrypted within a specified time (encryption throughput) and the amount of data/ciphertext that can be decrypted within a specified time (decryption throughput)) simulations I performed, on the average, Elgamal cryptosystem recorded 34821.52 b/s (bytes per second) in both the encryption of messages and decryption of ciphertext making it the one with the lowest total average throughput among the three cryptosystems used in the simulation test. RSA cryptosystem follows with a total average throughput of 1073651.38b/s (bytes per second) which is higher than Elgamal's a total average throughput but lesser than the proposed cryptosystem's total average throughput which is 2452253.1b/s (bytes per second), being the highest total average throughput compared to Elgamal and RSA cryptosystems' a total average throughputs. This clearly indicates that, Elgamal cryptosystem takes more computing resources (such as memory space and CPU time) in encrypting messages of any size and decrypting ciphertexts/messages of any size and therefore less efficient (i.e. can do less work/encrypt and decrypt less amount of data in a given amount of time), compared to the other two cryptosystems, whiles in contrast the proposed cryptosystem rather takes the least computing resources (such as memory space and CPU time) in encrypting messages of any size and decrypting ciphertexts/messages of any size and is therefore more efficient (i.e. can do more work/encrypt and decrypt more amount of data in a given amount of time) compared to Elgamal and RSA cryptosystems.

Therefore, if the priority is to encrypt messages or any size and decrypt ciphertexts of any size faster and efficiently thus to give higher encryption and decryption work output, then the best cryptosystem to choose is the Proposed Cryptosystem.

Figure 4.6 shows the average overall throughput comparison of the three cryptosystems.

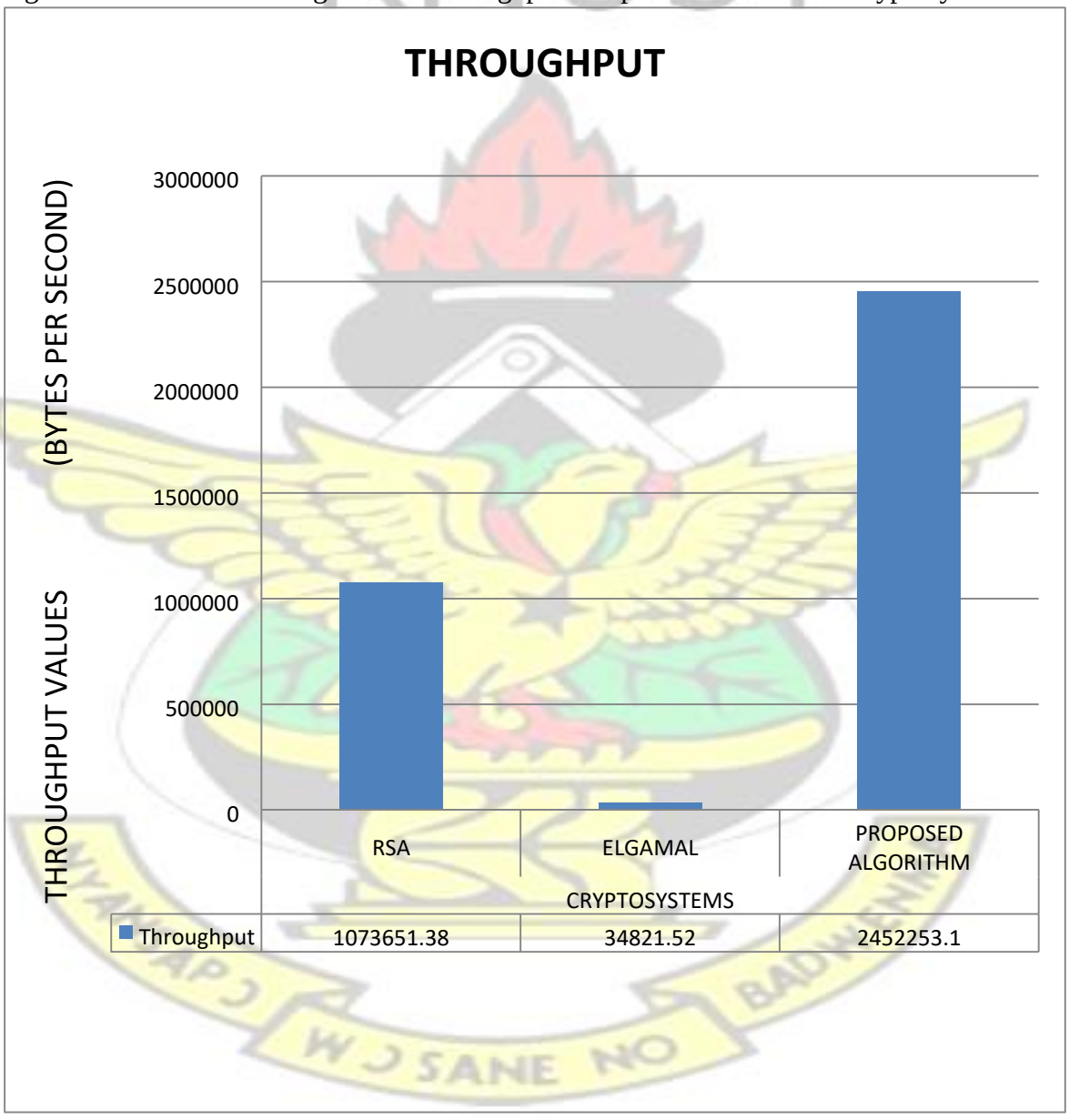


Figure 4.6 Cryptosystems and their average throughput in bytes/second

4.7 Randomness Comparisons

Taking a look at figure 4.7, it can be seen that for the property of the cryptosystems called randomness, all the cryptosystems under study were par for the fifty simulations I performed. This means all the cryptosystems under study could generate or produce random outputs for a particular given input or they could produce different cipher texts when the same plaintext is encrypted a lot of times. Thus any message or plain text that was simulated or passed through the cryptosystems were able to produce different cipher text no matter how many times the same message was fed into them. So for RSA, Elgamal and the Proposed Cryptosystems, a particular message encrypted 50 times was able to produce or generate 50 different outputs or cipher texts.

Therefore since one of the most desired properties of a cryptosystem is randomness and that also the strength of an asymmetric cryptosystem is proportional to the degree of randomness of the encrypted data, it therefore stands from the simulation analysis that all the three cryptosystems, thus RSA, Elgamal and the Proposed cryptosystems passed the test of randomness. They are therefore not deterministic and hackers will find it difficult to derive meaning of different cipher texts produced from the same plaintext. Therefore all three asymmetric cryptosystems passed and provides higher data security in terms of randomness of ciphertexts produced from the same plaintext.

KNUST

Figure 4.7 gives a chart of the randomness comparison of the three cryptosystems.

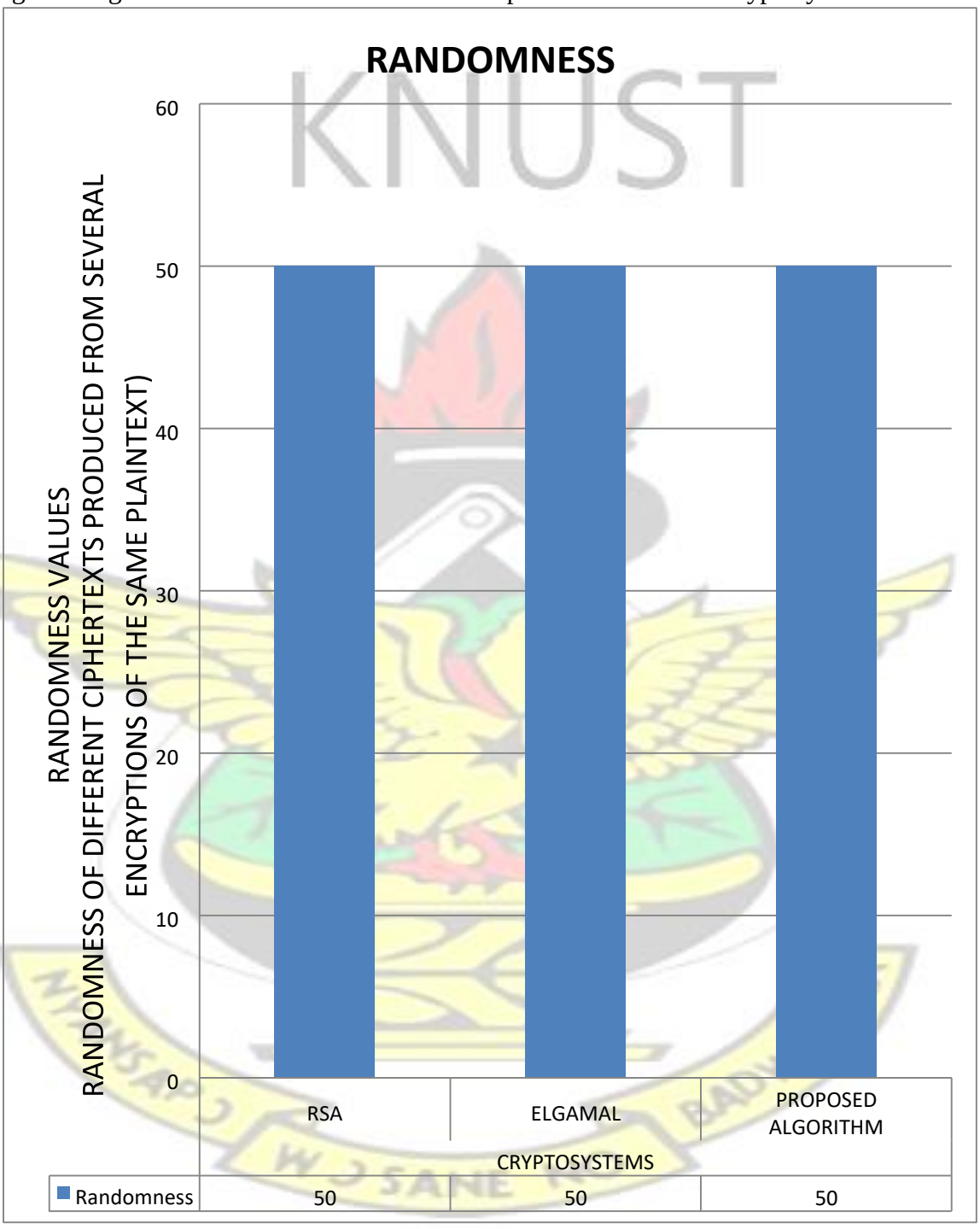


Figure 4.7 Cryptosystems and their average randomness

4.8 Key length Comparisons

With the public key sizes recorded, RSA had a key size of **1024** bits, Elgamal with**1280** bits, and **1280** bits key length for the Proposed cryptosystem. This shows that the proposed system and Elgamal cryptosystem recorded the highest public/encryption key lengths than the RSA cryptosystem. The strength of an asymmetric encryption algorithm is also directly proportional to the key size. Hence a higher key size (thus in this case the proposed cryptosystem and elgamal cryptosystem's public keys) results in an increase in data security of the cryptosystem, in the sense that, hackers who gets access to the public key and tries to use it to compute the private key in order to get the plaintext when they also have the cipher text, will find it very difficult when the key size is bigger. Moreover the general rule for asymmetric cryptosystems is that the longer the key the better the cryptosystem.

Multiple keys for the various cryptosystems were also considered and it was noticed that the encryption key or public key for RSA is made up of two (2) values, that is (e, n), Elgamal has three (3) values, that is (p, g, y) and the Proposed cryptosystem has four (4) values (y, x, s, t). Thus for the proposed cryptosystem the first public key (y, x) is synonymous to the e value in the public key of RSA and the proposed cryptosystem's second public key (s, t) represents the modulus n value in the public key of RSA. These set of values (y, x, s, t) are mathematically related and they are sent to the sender as two multiple public keys (y, t) and (s, x) seperately. Also for the proposed cryptosystem, because of the way the public key is created from large integer primes with their corresponding derived large modulus, it makes it difficult to factorize or find out the real

values of e and n from (y, t) and (s, x) or the decryption key d. Less knowledge on the mathematical relation of (y, t, s, x) by an attacker also contributes to the security of the cryptosystem. Thus, in terms of multiple key systems, the Proposed cryptosystem will be considered more secured followed by Elgamal then RSA cryptosystem.

Moreover from figure 4.8 cipher text length was also considered and from the fifty simulation tests I conducted, it was noticed that the Proposed cryptosystem produces a longer cipher text during encryption with an average value of 4004.16 bytes, followed by RSA with an average value of 2883.94 bytes then Elgamal cryptosystem with the least average value of 76.92 bytes. The longer the cipher text, the more it is considered to be secured because the text will be plenty and the attacker will have to do much work to figure out the plaintext. Based on this, the Proposed cryptosystem will be considered more secured followed by RSA, then Elgamal cryptosystem as seen in figure 4.8.

Figure 4.8 depicts the ciphertext length comparison of the three cryptosystems.

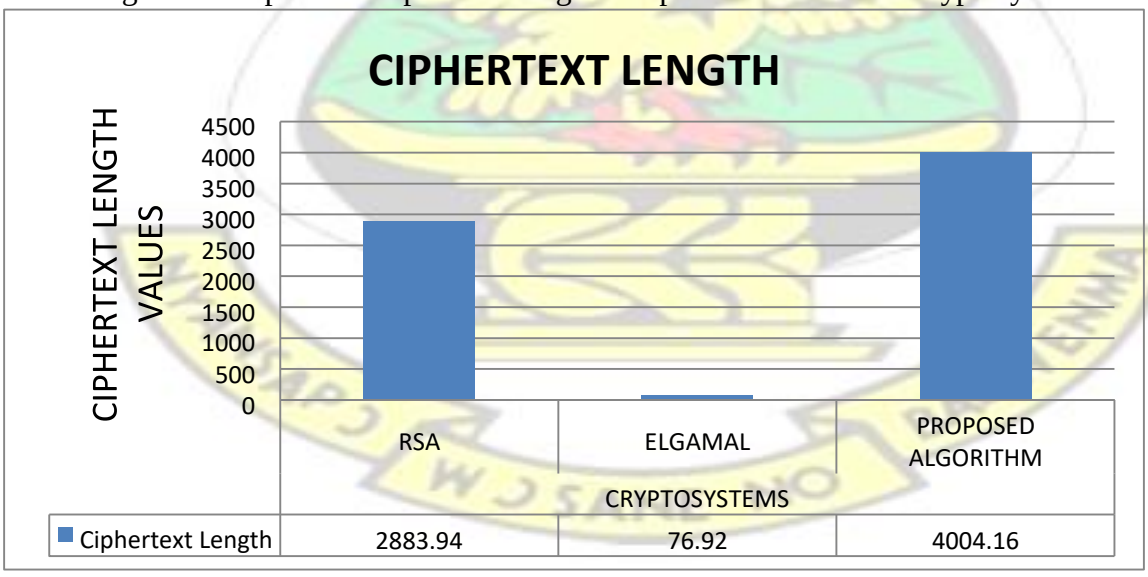


Figure 4.8 Cryptosystems and their average cipher text length

4.9 Operations per Instruction (O/I) Comparisons

Finally, the cryptosystems' property known as one operation per one instruction (O/I) (which predicts the difficulty to factor large prime numbers/modulus of the various cryptosystems) simulations was done fifty times and from figure 4.9, RSA recorded an average of 1.80058E-12 O/I which is the least among the three cryptosystems under study, followed by Elgamal which also recorded 2.58674E-12 O/I, and then with the Proposed Cryptosystem recording the highest operations per instruction value of 7.62205E-12 O/I. A higher value of O/I means a hacker will have to do a lot of work to be able to decrypt a message (one instruction) that is encrypted by a particular cryptosystem or defeat/break a particular cryptosystem. This therefore implies the Proposed cryptosystem will have the highest attack resistance level, thus the highest estimate of the amount of work and time that is required by hackers to defeat a cryptosystem. It is then followed by Elgamal and then RSA being the one with the least attack resistance level according to the simulation test analysis of figure 4.9.

Thus according to the O/I simulation test, the Proposed cryptosystem can provide the most security, followed by Elgamal then RSA cryptosystems.

KNUST

Figure 4.9 depicts the operation per instruction (i.e. the attack resistance level) comparison of the three asymmetric cryptosystems.

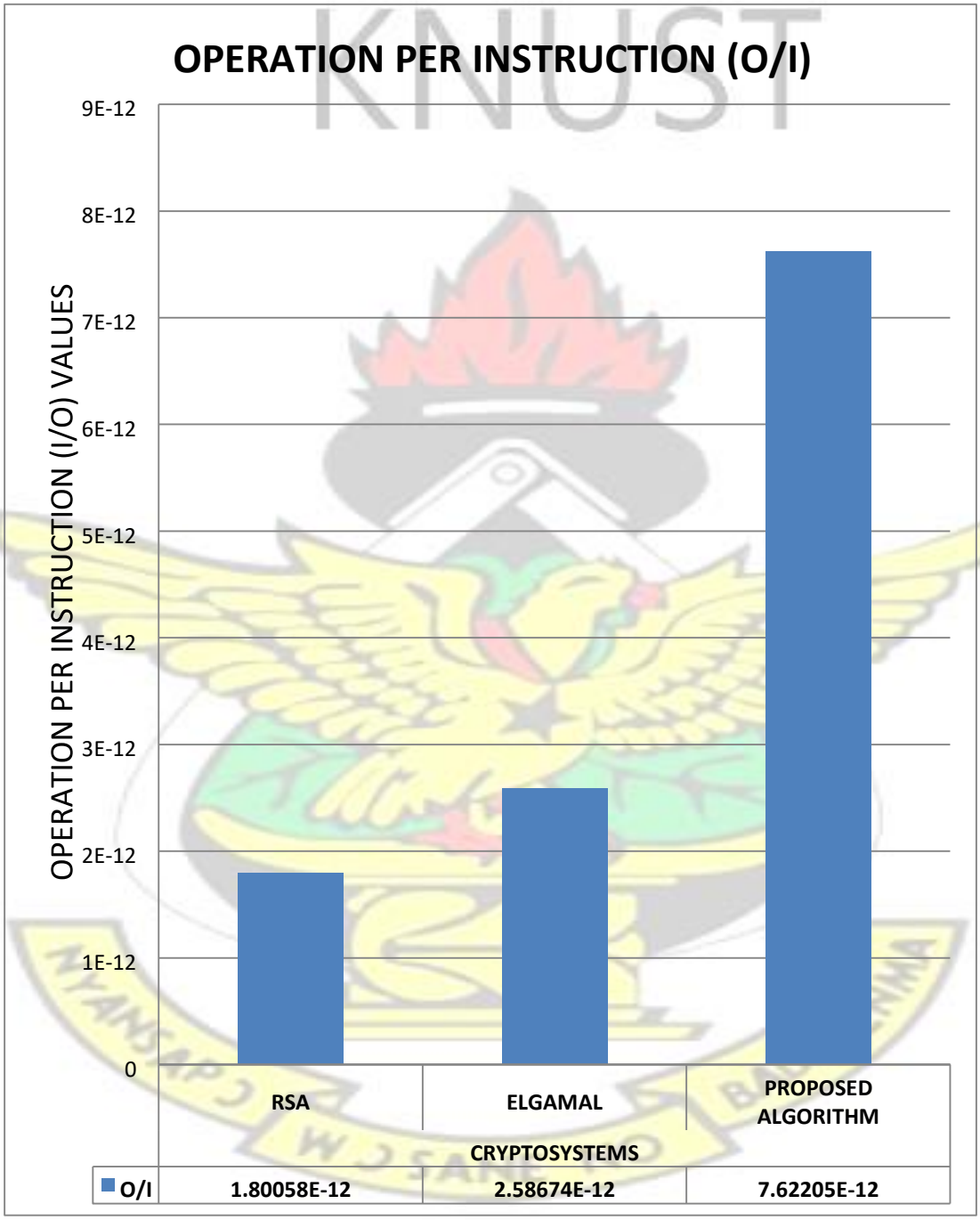


Figure 4.9 Cryptosystems and their average O/I

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The eminent characteristics of asymmetric cryptosystems are encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput, throughput, randomness, key length, Operation per Instruction (O/I).

Encryption computation time, decryption computation time and performance shows how optimized the cryptosystem is and how fast the cryptosystem can encrypt and/or decrypt a message. From the discussions made, it was observed that the proposed cryptosystem had better results in all these properties. This means that it is more optimized, consumes less computer resources and faster or uses less processing time to encrypt and/or decrypt a message. This is followed by RSA cryptosystem and then Elgamal cryptosystem.

From this paper/research also, it has been observed that, how productive and efficient a cryptosystem is, depends on the encryption throughput, decryption throughput and throughput of each algorithm. The larger the results for these properties, the better the cryptosystem in terms of work output. The proposed cryptosystem had better results in these properties. This indicates that the proposed cryptosystem is more productive and efficient and can encrypt and/or decrypt more messages in a short time. This is followed by RSA cryptosystem and then Elgamal cryptosystem.

Again, from the completed tests and research, it was noticed that, how secure a cryptosystem is, its attack resistance level and efficacy and efficiency is dependent on randomness, (key

length, cipher text length, multiple keys) and the Operation per Instruction (O/I) properties of the cryptosystem.

For randomness, all the cryptosystems were random.

For the key length, it is the same for the Proposed cryptosystem and Elgamal cryptosystem followed by RSA cryptosystem having the least key length. The cipher text length was longer in the proposed cryptosystem followed by RSA and then Elgamal. For multiple keys, the proposed cryptosystem has more keys, followed by Elgamal and then RSA.

Again for the Number of Operation per Instruction (O/I), the proposed cryptosystem requires more O/I followed by Elgamal and then RSA.

From these three properties it can be observed that, the proposed cryptosystem is more secured and has better key management followed by Elgamal and then RSA cryptosystem.

In all, the results of this research clearly demonstrate that, the proposed cryptosystem has better properties than the existing ones.

5.2 Recommendations

From the findings, analysis and conclusions of this research, I propose and recommend that the proposed cryptosystem should be the one used to encrypt and decrypt messages because is more secured, has better key management, efficient and faster when used.

REFERENCES

Alqdah, M. and Hui, Y. L. (2008). Simple Encryption and Decryption Application. *International Journal of Computer Science and Security, vol. 1(issue 1)*: pages 14-17, 33-44.

Andrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thom■, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Beguelin, S. and Zimmermann, P. (2015). Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. *22nd ACM Conference on Computer and Communications Security (CCS '15)*: pages 2-8.

Arya, P. K., Aswal, M. S. and Kumar, V. (2015). Comparative Study of Asymmetric Key Cryptographic Algorithms. *International Journal of Computer Science and Communication Networks, vol. 5(issue 1):* pages 17-21.

Ayele, A. and Sreenivasarao, V. (2013). A Modified RSA Encryption Technique Based on Multiple public keys. *International Journal of Innovative Research in Computer and Communication Engineering, vol. 1(issue 4):* pages 856- 863.

Bellare, M., Boldyreva, A. and Micali, S. (2000). Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. *Lecture Notes in Computer Science, vol. 1807, Springer-Verlag Berlin Heidelberg Publications:* pages 260-265.

Bhagat, P. V., Satpute, K. S. and Palekar, V. R. (2013). Reverse Encryption Algorithm: A Technique for Encryption & Decryption. *International Journal of Latest Trends in Engineering and Technology (IJLTET), vol. 2(issue 1):* pages 90-95.

Boneh, D. and Franklin, M. (2001). Efficient generation of shared RSA keys. *Journal of the ACM, vol. 48(issue 4):* pages 702-722.

Bu, S., Yu, F. R., Liu, X. P., Mason P. and Tang, H. (2011). Distributed Combined Authentication and Intrusion Detection with Data Fusion in High-Security Mobile Networks. *IEEE Transactions on Vehicular Technology, vol. 60(issue 3):* pages 1025-1036.

Charanjitsingh, G. (2013). Cryptography and its two Implementation Approaches. *International Journal of Innovative Research in Computer and Communication Engineering , vol. 1(issue 3):* pages 668-672.

Delfs, H. and Knebl, H. (2006).Introduction to Cryptography: Principles and Applications 2ed.. *Information Security and Cryptography; Texts and Monographs, Springer-Verlag Berlin Heidelberg Publications*: pages 41-73.

Elliot S.J., Peters J.L., Rishel, T.J. (2004). An Introduction to Biometrics Technology: Its Place in Technology Education. *Journal of Industrial Teacher Education (JITE), vol. 41(issue 4). http://scholar.lib.vt.edu/ejournals/JITE/v41n4/elliott.html*

Eschenauer, L. and Gligor, V. D. (2002). A Key Management Scheme for Distributed Sensor Networks. *ACM conference on Computer Security, vol.2:* pages 41-47.

Essays, UK. (November 2013). Advantages And Disadvantages Of Biometrics. Retrieved from *https://www.ukessays.com/dissertation/examples/information-systems/advantagesand-disadvantages-of-biometrics.php?cref=1.*

Farah, S., Javed, M. Y., Shamim, A. and Nawaz, T. (). An Experimental Study on

Performance Evaluation of Asymmetric Encryption Algorithms. *Recent Advances In Information Science.* Page 124.

Gambhir, A. (2014). RSA Algorithm or DES Algorithm. *Journal of Engineering Computers & Applied Sciences, vol 3(issue 4):* pages 27-28.

Gennaro, R. (2000). RSA-Based Undeniable Signatures. *Journal of Cryptology, vol. 13(issue 4):* pages 397-416.

Gordon, M. D., (1993). Discrete Logarithms in GF (p) Using the Number Field Sieve. *SIAM Journal on Discrete Mathematics, vol. 6(issue 1)*: pages 124-138.

Hankerson, D., Menezes, A., and Vanstone,S. (2004). Guide to Elliptic Curve Cryptography. *Springer –Verlag Berlin Heidelberg Publications:* pages 1-15.

Kahate, A. (2003). Cryptography and Network Security, 2nd ed., *Tata McGraw Hill (TMH)Publications:* pages 40-52.

Is The Future Here? Biometric Technology Pros and Cons!. *Biometric-Security-Devices.com*

Kakkar, A., Bansal, P. K. and Singh, M. L. (2012). Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network. *International Journal of Engineering and Technology (IJET), vol. 2(issue 1):* page 87-89.

Kaminsky, D. (2012). Survey is good, Thesis is strange. *http://www.dankaminsky.com*

Khalique, A., Singh, K., and Sood, S. (2010). A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards. *International Journal of Computer Applications, vol. 2(issue 3)*: pages 26-30.

Kim, H. W. and Lee, S. (2004). Design and Implementation of a Private and Public Key

Crypto Processor and Its Application to a Security System. *IEEE Transactions on Consumer Electronics, vol. 50(issue 1):* pages 214-224.

Koblitz, N., (1987). Elliptic Curve Cryptosystems. *Journal of Mathematics of Computation, published by American Mathematical Society, vol. 48(issue 177)*: pages 203-209.

Lenstra, A. K. and Lenstra, Jr. H. W. (1993). The Development of the Number Field Sieve. *Lecture Notes in Mathematics, vol. 1554, Springer-Verlag Berlin Heidelberg Publications:* pages 11-47

Lenstra, A. K., Hughes, J. P., Augier, M., Bos, J. W., Kleinjung, T. and Wachter, C., (2012). Ron was wrong, Whit is right, *EPFL IC LACAL*: pages 1-2.

Li, H. and Li, J. (2008). A New Compact Dual-Core Architecture for AES Encryption and Decryption. *IEEE Canadian Journal of Electrical and Computer Engineering, vol. 33(issue 3):* pages 209-213.

Li, J., H., Bhattacharjee, B., Yu, M. and Levy, R. (2008). A Scalable Key Management and Clustering Scheme for Wireless Adhoc and Sensor Networks. *Journal of Future Generation Computer Systems, Elsevier Science Publishers, vol. 24(issue 8):* pages 860869.

Li, Y., Ohta, K. and Sakiyama, K. (2012). New Fault-Based Side-Channel Attack Using Fault Sensitivity. *IEEE Transactions on Information Forensics and Security, vol. 7(issue 1)*: pages 88-97.

Mare, S. F., Vladutiu, M. and Prodan, L. (2011). Secret Data Communication System Using Steganography, AES and RSA.*IEEE 17th International Symposium for Design and Technology in Electronic Packaging:* pages 339-344.

Markoff, J. (2012). Flaw Found in an Online Encryption Method. *http://www.nytimes.com/technology.*

Meier, A.V (2005). The Elgamal Cryptosystem: pages 6-10. Millett,

L. I. and Holden, S. H. (2003). Authentication and its Privacy

Effects. *IEEE Internet Computing, vol. 7(issue 6):* pages 54-58.

Mohan, H. and Raji, R. (2011). Performance Analysis of AES and MARS Encryption Algorithms, *International Journal of Computer Science Issues, vol. 8(issue 4)*: pages 2-8.

Moore, S. K. (2012). UPDATE: RSA Responds to Flaw Finding. *http://www.spectrum.ieee.org/tech-talk.computing/it/rsa-flaw-found.*

Oded, G. (2004). Foundations of Cryptography: Basic Applications, vol. 2. *Cambridge University Press*: pages 10-19.

Orman, H. and Hoffman, P. (2004). Determining Strengths For Public Keys Used For Exchanging Symmetric Keys, *ISOC RFC 3766 (BCP 86):* pages 3-6

Parmar, K. and Jinwala, D. C. (2015). Symmetric-Key Based Homomorphic Primitives for End-to-End Secure Data Aggregation in Wireless Sensor Networks. *Journal of Information Security, vol.6 (issue 1):* page 39-41

Sakiyama, K., Li, Y., Ohta, K. and Iwamoto, M. (2012). Information Theoretic Approach to Optimal Differential Fault Analysis. *IEEE Transactions on Information Forensics and Security, vol. 2:* pages 109-120.

Seurin, Y. and Treger, J. (2013). A Robust and Plaintext-Aware Variant of Signed ElGamal Encryption. *Lecture Notes in Computer Science, vol. 7779:* pages 1-2.

Halevi, S. and Krawczyk, H. (1999). Public-key cryptography and password protocols. *ACM Transactions on Information and System Security, vol. 2(issue 3)*: pages 230-268.

Sharma, S., Yadav, J. S. and Sharma, P. (2012). Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2(issue 8):* pages 134-138.

Sheela, K. andRaj, E. G. D. P. (*2013*). Increased Key Size Method in SRNN Public Key Cryptography Algorithm. *International Journal of Computer Science and Mobile Computing, vol. 2(issue 8):* pages 219-223.

Singh, L. and Bharti,R. K. (2013). Comparative Perfomance Analysis Of Cyptographic Algorithms. *International Journal Of Advanced Research In Computer Science And Software Engineering (IJARCSSE), vol. 3(issue 11)*: page 11.

Singh, R. and Kumar, S. (2012). Elgamal's Algorithm in Cryptography, *International Journal of Scientific & Engineering Research, vol. 3(issue 12):* pages 1-4

Stallings,W. (2003).Cryptography and Network Security: Principles and Practice, 3rd ed.*Prentice Hall Publications*: pages 15-40.

Sun, H. M., Wu, M. E., Ting,W. C. and Hinek, M. J. (2007). Dual RSA and Its Security Analysis. *IEEE Transactions on Information Theory, vol. 53(issue 8):* pages 2922-

2933.

Taylor, G. and Cox, G. (2011). Behind Intel's New Random-Number Generator. *http://www.spectrum.ieee.org.*

Wan, Z., Liu,J. and Deng, R. H. (2012).HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing.*IEEE Transactionson Information Forensics and Security, vol. 7(issue 2):* pages 743-754.

Wang, K. (2009). An Encryption and Decryption Algorithm Implementation on FPGAs. *IEEE Fifth International Conference on Semantics, Knowledge and Grid, vol. 1*: pages 298-301.

Williams, H. C., (1980). A Modification of the RSA Public-Key Encryption Procedure. *IEEE Transactions on Information Theory, vol. 26(issue 6):* pages 726-729.

Yogita (2016). Analysis of RSA Encryption to Purpose Two – Step Improvement. *Imperial Journal of Interdisciplinary Research (IJIR), vol. 2(issue 6):* pages 641-642.