An Enhanced Asymmetric Cryptosystem using Multiple Key System

Steve Okyere-Gyamfi
Christian Service University College
Department of Computer Science and Information Technology
P. O. Box 3110, Kumasi, Ghana

J. B. Hayfron Acquah
Kwame Nkrumah University of Science and Technology (KNUST)
Department of Computer Science
Private Mail Bag, KNUST, Kumasi, Ghana

Vivian Akoto-Adjepong
University of Energy and Natural Resources
Department of Computer Science and Informatics
P.O. Box 214, Sunyani, Ghana

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 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 cryptosystems, which uses two different keys; one for encryption and the other for decryption, whiles symmetric cryptosystems use the same key for both encryption and decryption. The essential features of asymmetric cryptosystems that determines their efficiency and security are; encryption computation time, decryption computation time, performance, encryption throughput, decryption throughput, throughput, randomness, key length and Operation per Instruction (O/I). This research seeks to examine these properties of some asymmetric cryptosystems and subsequently develop a proposed cryptosystem that is more secure and efficient. The results of this research clearly demonstrate that, the proposed cryptosystem has better results for all the properties stated above.

General Terms

Security, Network, Cryptography, Cryptosystem, Asymmetric Cryptosystem, Algorithms, Key, Encrypt, Decrypt.

Keywords

Cryptography, Asymmetric Cryptosystems, RSA, Elgamal, Elliptic Curve, Encryption Computation Time and Throughput, Decryption Computation Time and Throughput, Performance, Throughput, Key Length, Multiple Keys, Randomness, Instruction per Operation, Private Key.

1. INTRODUCTION

The 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 [1].

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 information's confidentiality, integrity, authenticity, etc., most popularly called CIA triad [2]. Thus making security a very important element in network technology development that needs to be addressed, so that information can be protected from destruction and unauthorized intrusion. This problem of security has brought about the demand for the development of various technologies such as passwords, biometrics, patterns, cryptography, etc. to help eliminate information/network security issues, especially keeping information from unauthorized access [3].

But of these entire methods, one that has been known to be the best and also more reliable in keeping sensitive information confidential from unauthorized users is cryptography [4].

Cryptography is the study and practice of techniques that is used to make communication secure from intruders like hackers and attackers over networks. It makes the information unintelligible to a third party or an outsider by various ways of transformations. The presence/absence and type/number of keys determine the type of cryptosystem and also the encryption phase and decryption phase involved in that cryptosystem [3]. Depending on the number and kind of key(s) used, key based cryptosystems are classified or grouped into two. The first is Symmetric key cryptosystems like DES, AES, RC6, etc., which are usually called private/secret/conventional key cryptosystems), because of their usage of only one key for both encryption and decryption. Although, they are simple in their implementation yet their greatest vulnerability is the usage of only one key for encryption and decryption [5].The second type is Asymmetric key cryptosystems like RSA, Elgamal, etc., which are usually called public-key cryptosystems because the sender and receiver in this cryptosystem apply two different keys. Public/encryption key is used for encryption (convert plaintext/message to cipher text) and private/decryption key is used for decryption (convert cipher text to plaintext/message). The key for encryption is made public to all who would like to send message to the receiver (with whom the key for decryption is stored with in secret) [6].It is not possible to derive easily a corresponding secret key from a particular public key [7].

The following objectives will be addressed by my study:

- To analyze and study the various asymmetric cryptosystems used for data security and their various strengths and weaknesses.
- Develop a more secure and efficient asymmetric cryptographic model and compare the proposed approach with existing ones.
- Create awareness of the benefit of knowing and selecting an appropriate asymmetric cryptosystem.

2. REVIEW OF LITERATURE

Asymmetric cryptosystems and their variants can be put into three categories based on their design, namely: Integer Factorization model, Discrete Logarithm model and Elliptic Curve model cryptosystems.

Below is a discussion on all these three categories, the way they work, their strengths and weaknesses.

2.1 Integer Factorization Cryptosystem Models

Rivest et al in 1978 designed RSA cryptosystem at Massachusetts Institute of Technology (MIT) [3]. 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.

Two random prime numbers are used to generate public and secret keys.

For the RSA Key Generation Algorithm:

- 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.
- Compute the value n = p*q and (φ) phi = (p-1)(q-1)
- Choose a particular integer e, $1 < e < phi$, such that gcd (e, phi) = 1.
- Compute the exponent d, such that $1 < d < phi$, and that $ed \equiv 1 \pmod{phi}$.
- The public key then is (n, e) and private key is (n, d). The values d, p, q and phi are kept secret.
- The value n is called the modulus.
- The value e is called the public/encryption exponent.
- The value d is called the secret/decryption exponent.
- The key pair (n, e) forms the public key and it is made available to all or public.
- The key pair (n, d) forms the private/secret key and it is kept private [6].

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

The encryption steps are as follows:

- The sender gets the recipients public key.
- The message (M) or plaintext is first of all converted to a positive integer.
- The cipher text (C) is then computed.
- The cipher text is then sent by the sender
- 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;

- The private key is used for this process.
- Extracts plain text from cipher text which is represented as an integer.
- This is mathematically represented as:

$M= C^d \bmod n$ [5].

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 modification to RSA public-key cryptosystem suggests 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. However, the limitation/setback of this cryptosystem was the use of very large prime numbers which in turn brought about mathematical errors [8].

Another variant 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 [9].

A major strength 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 [7]. Yet RSA cryptosystem usually have low encryption speed [9]. Also 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 [10].

Over the years researchers have tried to check if there is a flaw in RSA, but all proved futile. But recently in a publication titled "Ron is Wrong, Whit is Right" [10], researchers conducted a comprehensive survey of public keys from the World Wide Web and one of their main target was to test (with the number sieve[18]) the properties of RSA cryptosystem and other public key algorithms based on Diffie-Helman. Their survey came to realize that out of every thousand keys, two may be vulnerable or not secure. Thus after collecting and examining about three (3) different sets of millions of openly accessible public keys, they realized that a very small percentage (about 1.1%) of these numbers were not actually random. Thus the result of this analysis showed that it is viable to find the private key or underlying numbers that was used to generate the public key.

RSA Company subsequently responded to this flaw finding of the survey in a publication by Moore that, the flaw is not from the algorithm itself but from its implementation. Thus according to RSA Security Company the flaw should be attributed to some of the hardware (i.e. embedded systems) and software used, which are not accurate and efficient enough to generate the random numbers [11]. Dan Kaminsky also attested to the fact that, the survey on RSA flaw was good in its survey but its thesis was wrongly done and also biased. 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 analyzed, were still able to have vulnerable keys then it means RSA have an advantage over the rest [12].

2.2 Discrete Logarithm Cryptosystem Models

Another major public key cryptosystem is the **Diffie–Hellman cryptosystem** which was published by Diffie and Hellman in 1976 and is based on the difficulty of computing the discrete logarithm theory.

Assuming the two parties K and L want to establish or settle on the encryption/ decryption keys to be shared using the key exchange algorithm of Diffie-Hellman; the steps below ensure:

- 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.
- K then selects a different large random integer j and computes b such that b=s*j mod r.
- K communicates the computed resulting number b to L.
- L also independently selects a different large random integer h and computes f such that f=s*h mod r.
- L also communicates the computed resulting number f to K.
- K then calculates the secret key Q1 as follows Q1= f*j mod r.
- 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, it can be transmitted offline between the two parties. Its use for only key exchange and also pre-computation of primes which may be improperly generated is its limitation [13].

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 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 [14][15].

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 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;

- He first generates/selects a random large prime integer (pim) which is usually 1024 bits.
- 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.
- He goes ahead and then generates/selects a random integer s which is also in the range $1\leq s \geq pim.$
- The public key/third part is then calculated as puk = ge^s mod pim.
- 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:

- Ama 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.
- In order for Ama 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;
- Thus for Ci1 and Ci2 computation:

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

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

- 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;

- Kofi has to calculate an inverse modular of $(Ci1)^{rn}$ modulo pim, represented as $(Ci1)^{-rn}$, and usually called the decryption factor.
- The original message which Ama encrypted is gotten through the following calculation M = Ci2 × $(Ci1)^{-rn}$ mod pim, thus for every block of message $n_{1,2,\ldots}$.=$(ge^s)^{-rn}$* Ci mod 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. [14]

Another shortcoming of the Elgamal cryptosystem is message expansion. This is so because there is doubling of the transferred message. 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[7]. Moreover according to Seurin and Treger who produced a variant of Elgamal, they noticed that Elgamal cryptosystem is vulnerable when it comes to adjustable attacks on certain cipher texts of it, because it is manipulative [15].

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. It is a little based on the discrete logarithm problem and covers a multiplicative finite fields group as shown by Fig. 1 [16]. ECC offers short encryption and decryption keys, and less power consumption [14]. Also ECC is applied in mostly resource constrained environments, like mobile networks and wireless ad-hoc networks [13].

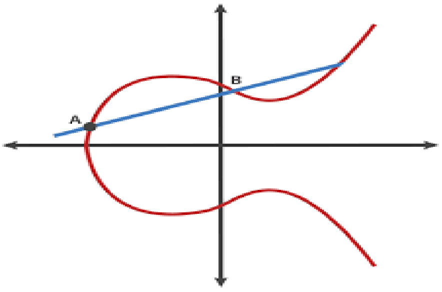

Fig. 1 Elliptic Curve Representation

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} and vice versa, the following steps with its subsequent computation is done.

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

The complexity of ECC's algorithm and cumbersomeness of its associated computations gives it a major drawback [17] [1].

Also out of the keys analyzed in the survey of Lenstra et al, 0.45 of them (which are those of ECC and Discrete Logarithm family) were duplicates with unauthorized ownership [10] and also a high cost of implementation are drawbacks of ECC [19].

3. METHODOLOGY

The proposed cryptosystem is based on RSA, it is more efficient and secure with good key management system. The proposed cryptosystem was designed, following the procedure below.

3.1 The Proposed Asymmetric Cryptosystem

The proposed cryptosystem is based on the model of RSA cryptosystem, but for the proposed 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 which are:

- Key Pair Generation
- Encryption Process
- 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;

- Choose 3 large primes, a, b, c
- Calculate n as n=a*b*c
- Represent n as t/s.

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

- The value of e should be greater than 1 and less than z where: $[z = (a - 1)\ (b - 1)\ (c - 1)]$
- 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.)
- 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; 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.

It will be difficult to find out the values of e as it is represented with the values (y, t) and (s, x), 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 cryptosystem in terms of data security.

The forming of the private key is also done as follows:

- The calculation of the private key d is done using the public key e and z. For a particular z and e, there exists a unique value d.
- 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 mod z
- 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 is performed by the intended sender (Kofi) as follows:

- 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).
- The sender (Kofi) converts the plaintext P into a series of numbers that is less than n.
- In order to encrypt this plaintext P, the simple mathematical step below is used;

(Cipher text)C = P $^{x/y}$ mod (t/s)

- 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 is performed by the intended recipient (Ama) of the plaintext (P) as follows:

- The process of decryption is very straightforward. Suppose the receiver (Ama) of plaintext (P) has received its cipher text (C).
- The receiver (Ama) raises C to the power of his private key (d). The result modulo n then becomes the plaintext (P). Mathematically written as: Plaintext (P) = C ^{d} mod n

Fig 2 depicts flowchart of the proposed cryptosystem.

4. IMPLEMENTATION

An experimental research strategy and a quantitative approach were used in this research. A test suite was used to check and evaluate the proposed cryptosystem and the popular existing ones such as RSA and Elgamal properties. The properties that were used for the evaluation was Encryption computation time, Decryption computation time, Performance, Encryption throughput, Decryption throughput, Throughput, Randomness, Key length and Operation per instruction (O/I). Results were recorded for further analysis after running data 50 times on the test suite [3][7][13][1][18].

5. ANALYSIS AND DISCUSSION

Analysis and discussion was done by considering the various cryptosystems and the results of their properties.

5.1 Encryption Computation Time Comparisons

Considering Fig. 3, for the fifty simulations performed, on the average, it took 5963215.84 nanoseconds to encrypt messages using Elgamal, and is the highest among the cryptosystems. RSA follows with 3960663.58 nanoseconds and the proposed cryptosystem used 2369631.76 nanoseconds which is the least.

Moreover considering Fig. 4 & 5, the three cryptosystems were used to encrypt three groups of text (which are grouped according to their text sizes); it was seen that Elgamal cryptosystem still came out with the highest values for all the three groups of different sizes of text encrypted, thus 1765784.4 nanoseconds, 4784357.9 nanoseconds and 231000580.9 nanoseconds for small text, medium text and large text sizes respectively. RSA once again followed with values 691641.7 nanoseconds, 3620585.6 nanoseconds and 164377105.7 nanoseconds for small text, medium text and large text sizes respectively and the proposed cryptosystem still having the least values of 307847.3 nanoseconds, 1531408.6 nanoseconds and 103418651.7 nanoseconds for small text, medium text and large text sizes respectively.

Therefore, if the priority is to encrypt messages of any size faster, then the best cryptosystem to choose is the Proposed Cryptosystem followed by RSA and then Elgamal cryptosystem.

5.2 Decryption Computation Time Comparisons

Taking Fig. 3 into consideration, for the fifty simulations performed, on the average, it took 9936733.4 nanoseconds to decrypt ciphertext using Elgamal and it is the highest among the cryptosystems, RSA follows with 2098261.82 nanoseconds and the proposed cryptosystem recorded 1442936.32 nanoseconds, which is the least.

Moreover considering Fig. 4 & 5, when the three cryptosystems were used to decrypt three groups of cipher text (which are grouped according to the text sizes), it was realized that, Elgamal cryptosystem still came out with the highest values for all the three groups of different sizes of ciphertexts decrypted with values: 3105562.4 nanoseconds, 7670060.7 nanoseconds and 573076601 nanoseconds for small text, medium text and large text sizes respectively, RSA once again following with values: 152518 nanoseconds, 1999398.6 nanoseconds and 123330012.4 nanoseconds for small text, medium text and large text sizes respectively and the proposed cryptosystem once again recorded the least values: 96963.6 nanoseconds, 1165448.9 nanoseconds and 103983884 nanoseconds for small text, medium text and large text sizes respectively.

Therefore, if the priority is to decrypt ciphertext of any size faster, then the best cryptosystem to choose is the Proposed Cryptosystem followed by RSA and then Elgamal cryptosystem.

5.3 Performance or (Encryption and Decryption Speed) Comparisons

Considering Fig. 3, Fig. 4 and Fig. 5 for the fifty simulations performed, the total average performance of Elgamal was 16101573.24 nanoseconds which is the highest value, RSA follows with a 6058925.4 nanoseconds and the proposed cryptosystem performance value was 3812567.6 nanoseconds which is the least.

Therefore, if the priority is to encrypt messages of any size and decrypt cipher texts of any size faster and efficiently, then the best cryptosystem to choose is the proposed cryptosystem.

5.4 Encryption Throughput Comparisons

Looking at Fig. 6, for the fifty encryption throughput (i.e. the amount of data/messages that can be encrypted within a specified time) simulations performed, on the average, Elgamal recorded the lowest value of 18897.58 bytes per second, RSA cryptosystem follows with 996992.3 bytes per second and the highest value 2241522 bytes per second was recorded for the proposed cryptosystem.

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.

5.5 Decryption Throughput Comparisons

Considering Fig 6, for the fifty decryption throughput (i.e. the amount of data/ciphertext that can be decrypted within a specified time) simulations performed, on the average, Elgamal recorded the lowest value of 15917.8 bytes per second, RSA follows with 78330.06 bytes per second and the proposed cryptosystem recorded the highest value of 84029.26 bytes per second.

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.

5.6 Throughput Comparisons

Considering Fig. 6, for the fifty total throughput (i.e. encryption throughput and decryption throughput) simulations performed, on the average, Elgamal recorded the lowest value of 34821.52 bytes per second, RSA follows with 1073651.38 bytes per second and the proposed cryptosystem recorded the highest value of 2452253.1bytes per second.

Therefore, if the priority is to encrypt messages of any size and decrypt cipher texts 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.

5.7 Randomness Comparisons

Taking a look at Fig. 7, it can be seen that for randomness property of the various cryptosystems, all the cryptosystems under study were par for the fifty simulations performed. This means all the cryptosystems under study could generate random outputs for a particular given input or they could produce different cipher texts when the same plaintext is encrypted at different times. Thus any plain text that was encrypted or that passed through the cryptosystems were able to produce different cipher text no matter how many times the same message was fed into them.

Therefore since one of the most desired properties of a cryptosystem is randomness and that, 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, 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.

5.8 Key Length Comparisons

With the public key sizes recorded, RSA had a key size of 1024 bits, Elgamal with1280 bits, and 1280 bits key length for the proposed cryptosystem. This shows that the proposed system and Elgamal cryptosystem recorded the highest public key lengths than the RSA. The strength of an asymmetric encryption algorithm is also directly proportional to the key size. Hence a higher key size 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 have access to 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 value of (x/y) is synonymous to the e value in the public key of RSA and the value of (t/s) 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) separately.

5.9 Operations per Instruction (O/I) Comparisons

Finally, for one operation per one instruction (O/I) property of cryptosystems' (which predicts the difficulty to factor large prime numbers/modulus of the various cryptosystems), simulations was done fifty times and from Fig. 8, RSA recorded the least value of 1.80058E-12 O/I, followed by Elgamal which also recorded 2.58674E-12 O/I, and then 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 ciphertext (one instruction). 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 the cryptosystem

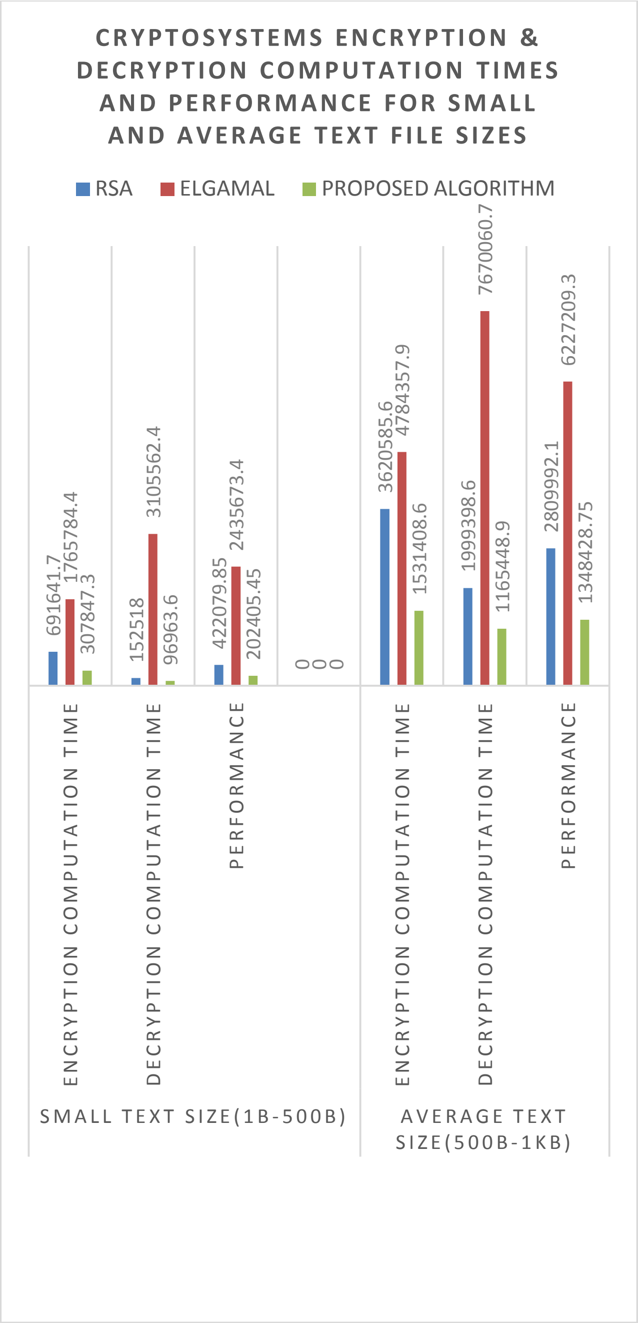

Fig. 5 Cryptosystems and their average encryption & decryption computation times and performance in nanoseconds for Large text sizes

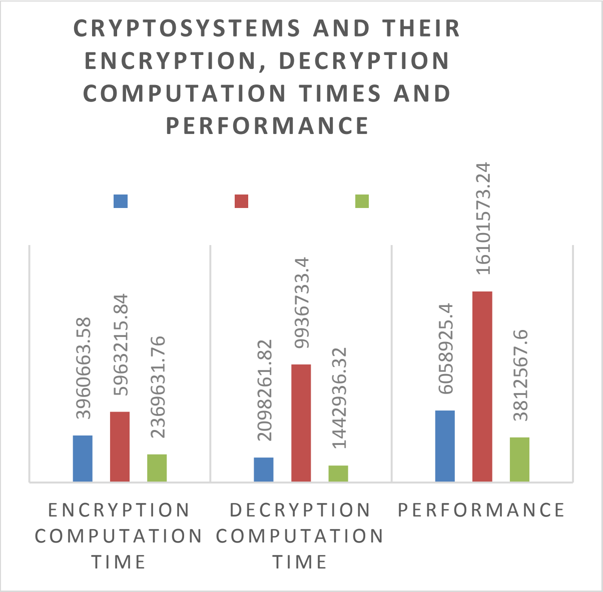

Fig. 3 Cryptosystems and their average encryption & decryption computation times and performance in nanoseconds

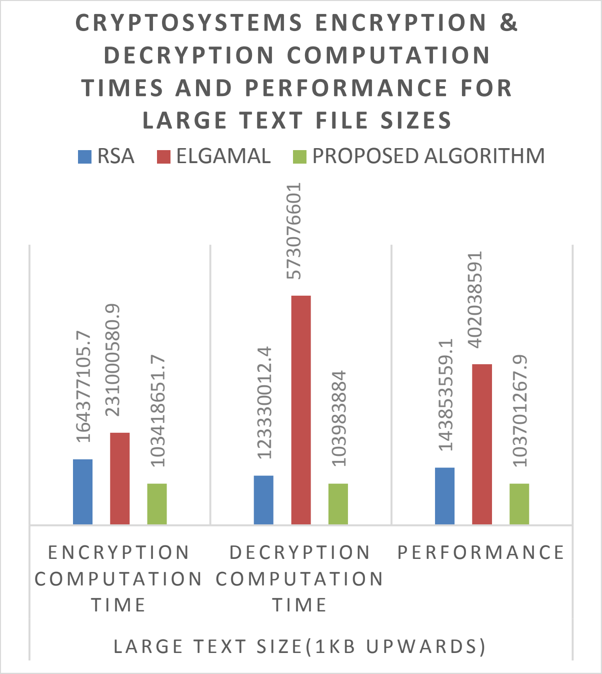

Fig. 4 Cryptosystems and their average encryption & decryption computation times and performance in nanoseconds for Average and Small text sizes

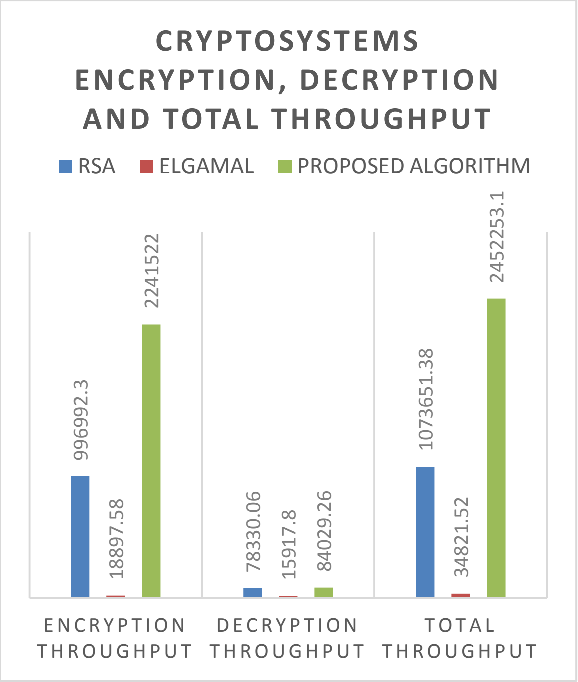

Fig. 6 Cryptosystems and their average encryption, decryption and total throughput in bytes per second

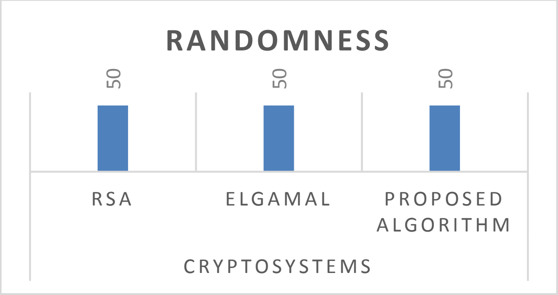

Fig. 7 Cryptosystems and their average randomness

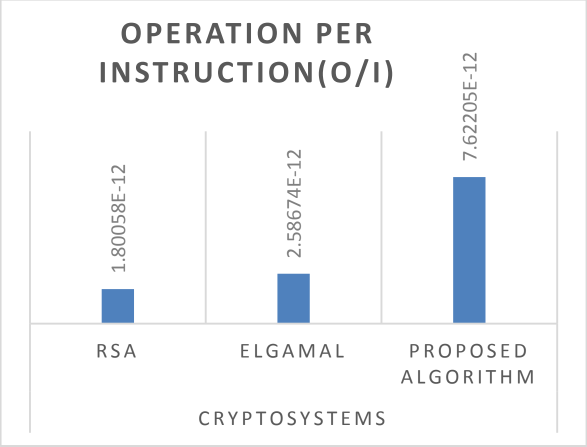

Fig. 8 Cryptosystems and their Operations per Instruction (O/I)

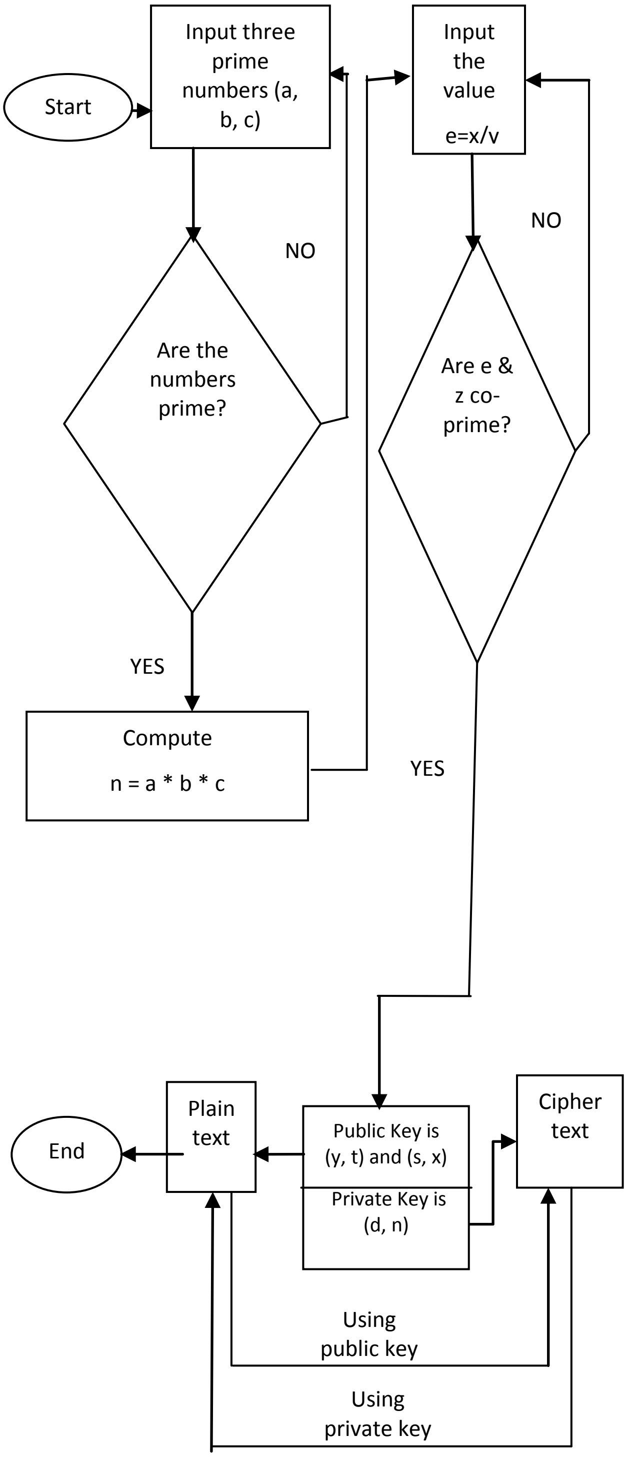

Fig. 2 Flowchart of Proposed Cryptosystem

6. CONCLUSION

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 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, efficacy and efficiency is dependent on randomness, key 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. 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.

7. ACKNOWLEDGMENTS

We express our sincere gratitude to almighty God and to all who made this paper a success. God bless you.

8. REFERENCES

[1] 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): pp. 11.

[2] Gambhir, A. (2014), “RSA Algorithm or DES Algorithm,” *Journal of Engineering Computers & Applied Sciences,* vol 3(issue 4): pp. 27-28.

[3] 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): pp. 87-89.

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

[5] 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): pp. 214-224.

[6] 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): pp. 134-138.

[7] Orman, H., and Hoffman, P., (2004). “Determining Strengths For Public Keys Used For Exchanging Symmetric Keys,” ISOC RFC 3766 (BCP 86): pp. 3-6.

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

[9] 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): pp. 2922-2933.

[10] 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: pp. 1-2.

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

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

[13] 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): pp. 17-21.

[14] Singh, R., and Kumar, S., (2012). “Elgamal‘s Algorithm in Cryptography,” *International Journal of Scientific & Engineering Research,* vol. 3(issue 12): pp. 1-4.

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

[16] Hankerson, D., Menezes, A., and Vanstone, S., (2004). *Guide to Elliptic Curve Cryptography*. Verlag Berlin HeidelbergPublications: Springer, pp: 1-15.

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

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

[19] 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): pp. 39-41.