1. **Introduction**

This chapter would just briefly look into the introduction of the requirements needed for the project and its purpose and aim. Also, a simple development plan for the prototype system which was drafted out is being presented here in this chapter. An overview of the system initially planned to be developed is also being presented here.

As mentioned earlier RSA is one of the most popular encryption algorithm that was devised. MREA was a further advancement in the RSA algorithm which increased the strength of the security that RSA was providing the users with. On the other hand MREA took more time to encrypt and decrypt data.

The main objective of this project is to devise a time efficient strong cryptographic algorithm. The main feature of this proposed system will be that it will be more secure compared to its counterparts and less time consuming than the MREA algorithm. The aim of the project is to assure secure transmission of data over a network without a third party intervention or in other words the messages sent would only be understood by the sender and to the intended receiver. That is, it would be difficult for a third party to understand the message even if he/she the cipher text. Apart from the security point of view the other main domain which our project aims is the time taken by the algorithms to compute the key generation, encryption and decryption calculations. The proposed system will try to decrease the time taken by the MREA algorithm.

**1.1 Motivation**

The RSA and MREA encryption algorithm which is quite a reliable algorithm for security purposes even till today, there are certain areas where they algorithm falls behind. But as time has passed RSA has shown its own drawback and so MREA was devised. This Algorithm increases the security of transmission of data as compared to RSA the time taken by MREA is large and so the level of security it provides is negligible for any application. So it was necessary to develop a new intermediate algorithm.

* 1. **Problem Statement**

The RSA and the MREA algorithm are the two existing algorithms that have been considered for this proposed project. MREA algorithm makes use of four prime numbers for key generation. Also the algorithm has some additional number of variables like µ, λ, g whose computation again require time. The Time Complexity of RSA is O (k3) whereas that of MREA is O (k3+k), this extra computation time is because MREA makes use of many keys instead of just two and also includes a large amount of exponent computation. Because of the above mentioned computations the simulation results shows that MREA takes six times moretime for the entire process of encryption and decryption than that of original RSA

* 1. **Scope of the Project**

The Scope of this project is to increase the time efficiency of MREA by using ideas such as Offline Storage and BigInteger library. In Offline Storage method the system will compute all the keys required for the algorithm before the encryption process and store in a database so as to saves the computation time for key generation for the next encryption process.

1. **Literature Survey**

Several encryption algorithms were being studied that were been researched and implemented by other researchers. For further understanding on other methods and techniques, refer to the reference page at the end of this report to search for the published papers.

The first paper is on the famous encrypting RSA algorithm which basically deals with the RSA algorithm and its merits and demerits. The second paper is on the MREA algorithm which is basically the modification on the RSA algorithm and it uses four prime numbers for key generation. Also it has many keys as compared to only two keys in RSA the encryption and decryption. These in turn increases the security of the algorithm but on the other hand the time required for encryption and decryption is increased. The time complexity is six times more than that of RSA algorithm.

The next paper is based on using the offline Storage on RSA algorithm with the purpose of decreasing the time required by the RSA algorithm. This strategy aims at computing all the keys required by the algorithm and storing then in tables prior to the commencement of the encryption process so that when the actual process starts the system can retrieve the keys and speed up the process of encryption and decryption. Thus the time required for the key generation can be saved and the algorithm works faster than normal RSA.

BigInteger Library is basically a library which has lot of predefined arithmetic functions. BigInteger library suggests that the use of these functions to compute the cipher text will speed up the process of encryption and decryption thereby reducing the Time complexity of the RSA algorithm. These functions include modulus, addition, subtraction, power functions and many more functions.

**2.1 EXISTING SYSTEM**

There are many encrypting algorithms which help in transmitting secured information around an in secured network. Following are already existing algorithms that has been considered for this project

* 1. **RSA Algorithm**

Named after its inventor RSA encryption algorithm is one of the most popular algorithm used for the security of the information sent across a network. It is a public key or asymmetric key cryptography which basically uses two prime numbers for its key generation. Two keys are generated which are public 'e' and private 'd' keys which are related to each other. The strength of the RSA lies in the fact that it is very hard to find factors for large numbers. Even a powerful hacking attack like brute force will also take a lot number of tries in order to get the right numbers used for key generation. The encryption and the decryption of the RSA takes only less numbers of modulus and the keys generated for the same are also very less as compared to others.

**RSA Algorithm :**

i. Select two prime numbers p and q.

ii. Find n=p\*q, Where n is the modulus that is made public. The length of n is as the considered RSA key length.

iii. Choose a random number ‘e’ as a public key in the range 0<e<(p-1)(q-1) such that gcd(e,(p-1)(q-1))=1.

iv. Find private key d such that ed\_1(mod (p-1)(q-1)).

**Encryption**

i. Consider the device A that needs to send a message to B securely.

ii. Let e be B’s public key. Since e is public, A has access to e.

iii. To encrypt the message M,represent the message as an integer in the range 0<M<n.

iv. Cipher text C = Me mod n, where n is the modulus.

**Decryption**

i. Let C be the cipher text received from A.

ii. Calculate Message M = Ce mod n, where d is B’s private key and n is the modulus.

* 1. **Modified RSA Encryption Algorithm (MREA)**

To increase the security of the existing RSA a Modified version of the RSA was being devised. Modified algorithm introduced additional keys so as to encrypt and decrypt the data. It also introduced lot of modulus operation which makes the computation much more complex but the security of the entire system is increased to great extentu. Because if it was difficult to find two factors of the a particular number then it becomes a lot more difficult to compute four factors. So thus MREA provides better security as compared to RSA.

**MREA Algorithm :**

**A. Key Generation Algorithm:**

i. Choose four large prime numbers p, q, r and s randomly and independently of each other. All primes should be of equivalent length.

ii. Compute n = p x q, m= r x s, phi = (p-1) x (q-1) and λ=(r-1) x (s-1).

iii. Choose an integer *e*, 1 < e <phi, such that gcd (e,phi) = 1.

iv. Compute the secret exponent *d*, 1 < d <phi, such that e x d mod phi=1.

v. Select an integer g=m+1.

vi. Compute the modular multiplicative inverse: µ = µ-1 mod m.

vii. The public (encryption) key is (n, m, g, e).

viii The private (decryption) key is (d, µ, λ).

**B. Encryption:**

i. Let m be a message to be encrypted where 0<mesg< n.

ii. Select random r where r < m.

iii. Compute cipher text as: c=g mesg^emodnxrm mod m2

**C. Decryption**

i. Compute message: m = (((cλ mod m2 – 1)/ m) xµmod m)dmod n.

1. **SOFTWARE REQUIREMENT SPECIFICATION**

# 3.1. Introduction.

# 3.1.1 Purpose

The purpose of the **JNS Encryption Algorithm** is to provide the following :

**Authentication**– Providing ones identity before granting access.

**Privacy & Confidentiality**– Ensure that outsiders cannot read data intended for specific parties.

***Integrity*** – Ensures that the message is not modified in any way before it arrives to intended recipient.

**Non-Repudiation**– Ensuring that the message truly originates from the sender.

## 3.1.2 Scope

The scope of the project is to provide faster encryption and decryption with the help of the proposed algorithm using Offline storage and Big Integer Library.

The comparative analysis of the JNS algorithm, RSA algorithm and the MREA algorithm based on their time complexity and security is done and relative graphs are generated to portray the results.

**3.1.3 Overview**

This Software Requirements Specification (SRS) is the requirements work product that formally specifies Encryption and Decryption using **JNS** Encryption Algorithm. Requirements statements in this document are both functional and non-functional.

These include a general description of the product, user characteristics, and general constraints of this system.

This model demonstrates the development team's understanding of the business domain and serves to maximize the team's ability to build a system that truly does support the business.

These requirements will allow the user to gain knowledge, as to how different encryption algorithms differ from each other in terms of providing security.

**3.2 Definitions, Acronyms, and Abbreviations**

**3.2.1 Definitions**

**Encryption:**

In cryptology, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor. In an encryption scheme, the message or information, referred to as [plaintext](http://en.wikipedia.org/wiki/Plaintext), is encrypted using an encryption algorithm, generating [cipher text](http://en.wikipedia.org/wiki/Ciphertext) that can only be read if decrypted.

**Decryption:**

The process of converting cipher text (unreadable form) back to plaintext is called decryption.

**3.2.2 Acronyms and Abbreviations**

**MREA:**

Modified RSA Encryption Algorithm

**RSA:**

Famous encryption algorithm named after its inventors Ron Rivets, Adi Shamir, and Leonard Adelman.

**JNS:**The researched new algorithm proposed in the project.

**Actor:**

actor

A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems.

**Use Case:**

http://res.dotnetcoders.com/images/uml/usecase-usecase.png

Use Cases describes the behavior of the system as seen from an actor’s point of view .Use Cases are depicted with an ellipse.

|  |
| --- |
|  |

**System boundary:**

<<subsystem>>

A system boundary defines the scope of what a system will be. A system cannot have infinite functionality . A system boundary of a use case diagram defines the limits of the system. The system boundary is represented with a rectangle diagram.

**Association:**

Associations are used to link Actors with Use Cases, and indicate that an Actor participates in the Use Case in some form. Associations are depicted by a line connecting the Actor and the Use Case.

**Aggregation:**

A special form of association that is used to represent a wide range of connections among set of object is denoted by a simple line with a diamond at the container end of the association.

**Generalization:**

Generalization is used when you find two or more use cases that have commonalities in behavior, structure, and purpose. When this happens, you can describe the shared parts in a new, often abstract, use case that is then specialized by child use cases.

**Include:**

*<<include>>*

It uses to include use cases that exhibit additional features.

**Extend:**

*<<extend>>*

It is used for specifying exceptional behavior.

**Class:**

Name

Attribute

Operations

Each class is a collection of data and functions that manipulate the data. Each class is represented by rectangle sub-divided into 3 categories followed by name, attributes and operations.

**Compositions:**

It is a strong form of association. The lifetime of the part is dependent upon the whole.

**Object:**

Objects are entities that encapsulate state and behavior. Each object has an entity and is distinguishable from other objects.

**Lifeline:**

Lifeline is a time span of the object. They are vertical dash line that represents the existence of an object at a particular time.

**Focus of control:**

Active procedure on which object is active. It is depicted by vertical rectangle.

**Types of message:**

***Synchronous:***

The sender waits until the responder finishes.

***Asynchronous:***

The sender does not wait for anything from the responder, but it continues its own activity.

***Flat:***

The sender does not wait for anything from the responder and finishes its activity, the control is then passed to the responder.

**State:**

A simple state with no substructure.

**Initial Node:**

Start state. It indicates the starting state of the system.

**Final Node:**

Final State. Indicates the end of the activity

**Fork/Join:**

Fork represents that many activities can be carried out in parallel.

**Swim lanes:**

Diagrams are partitioned according to the class who is responsible for carrying out the activities. They are represented as rectangles enclosing a group of action

**3.3. Overall Description**

**3.3.1 Product Perspective**

The software will run as a standalone application.

While the software is running, the user shall be able to select which component to input values. This incorporates the selection of any plaintext producing its resultant cipher text

**3.3.1.1 System Interfaces**

The system is intended to interface with a single user at any given time. The user is able to interact with the system using a graphical user interface.

**3.2.1.2 User Interfaces**

The user interface must provide the user an understandable and effective way for entering attributes into the system. Java will be used to create the graphical user interface for the system. At the top of the GUI is a drop-down panel that will display each of the encryption algorithms that may be used (including the RSA algorithm, MREA algorithm & the JNS Encryption algorithm). Below that is a text field that allows the user to enter the plaintext that is to be encrypted. Depending on the type of algorithm chosen, the user will also be given the option of choosing various options such as the various key or substitutions. At the bottom of the interface is a text area that displays the result of the cryptographic scheme or cipher text. Occupying the right side is a console area that displays any output that will attempt to give the user some insight into how the encryption schemes work.

**3.3.2 Product Function**

Encryption converts the input data in the garbled form so that no intruder can understand that form of data unless he/she is in possession of the secret key. Decryption converts the encoded message back to the plaintext.

Generation of graphs based on the performance of the RSA algorithm, MREA algorithm and JNS algorithm.

**3.3.3 General Constraints:**

**Data**

The maximum input block size is 128 bits.

The key size can be either 128,192 or 256 bit only.

**Language**:

The application and the algorithm will be designed using JAVA language.’ MySql’ database is used to store the data

**3.4. Specific Requirements**

**3.4.1 External Interface Requirements**

**3.4.1.1 User Interfaces**

Keyboard

Mouse

Display devices

File browser

Applet viewer

**3.4.1.2 Encryption Algorithms**

The system must be able to allow the user to select from the following encryption schemes:

**RSA:** RSA is a public key cryptographic algorithm that uses two prime number to generate the public as well as private key

**MREA:** A modified RSA algorithm which increases the security of the existing RSA algorithm by generating more number of keys for encryption and decryption.

**JNS:** The proposed algorithm that can be applied for practical use. It makes use o offline storage and BigInteger library so as to give better security and to reduce the time complexity.

**3.4.2 Functional Requirements**

**3.4.2.1 Viewing Messages**

**3.4.2.1.1 Plaintext**

The plaintext message is viewed that the user shall input.

**3.4.2.1.2 Cipher text**

The encoded cipher text is viewed by the user in one of a variety of forms that are used in encodings today, mainly the ASCII standard, decimal, hexadecimal, and binary.

**3.4.2.2 View Console**

The messages will be displayed to the user which will give them an insight into the inner workings of the algorithm.

**3.4.3 Non-Functional Requirements**

**3.4.3.1 User Interface and Human Factors**

Though the user shall access this system via a standalone application, the system will still function independently. All input fields will be aligned in the center of any panel that contains the component for ease of use reasons.

**3.4.3.2 Deployment Platform**

**3.4.3.2.1 Software**

Since the system runs as a standalone application, the user must have a recent version of Java2 installed on their system. Also the system should have all libraries and files for smooth functioning of ‘MySQL database’

**3.4.3.2.1.1 Operating System**

Because Java is used, the operating system will be irrelevant. The system will run on any OS that supports Java2, including, but not limited to, Windows, Linux, and MacOS. Also the system also requires a java virtual machine(jvm) for the algorithm to be executed.

**3.4.3.2.1.2 Graphics**

A graphical user interface is used to enhance usability and must be able to run on display that supports 640x480 resolution or higher

**3.4.3.2.2 Hardware**

The system must run on all systems that support Java2.

**3.4.3.3 Error Handling**

The system must check to ensure that all values in fields are valid (if an input field expects a letter of the alphabet, then great pains must be taken to ensure that the user does not input a non-alphabetic character), and that numerical fields have values that lie within the specified range.

**3.4.3.4 Quality Issues**

The program should not halt or crash during execution. The program must not alter the state of the user’s system after closing. As far as security issues are concerned, no files on the user’s system are read or otherwise modified.

**3.4.4 Software System Characteristics**

* **Reliability**

Since the algorithm used is much faster as compared to the original AES algorithm and since there are no known security flaws yet, the reliability of this software is high.

* **Availability**

The software is available to any party wishing to access it and will be accessible via the products home page

* **Security**

There are no known security flaws at this time.

* **Maintainability**

The source code should be properly documented, so that new developers will be able to understand the code as easily as possible

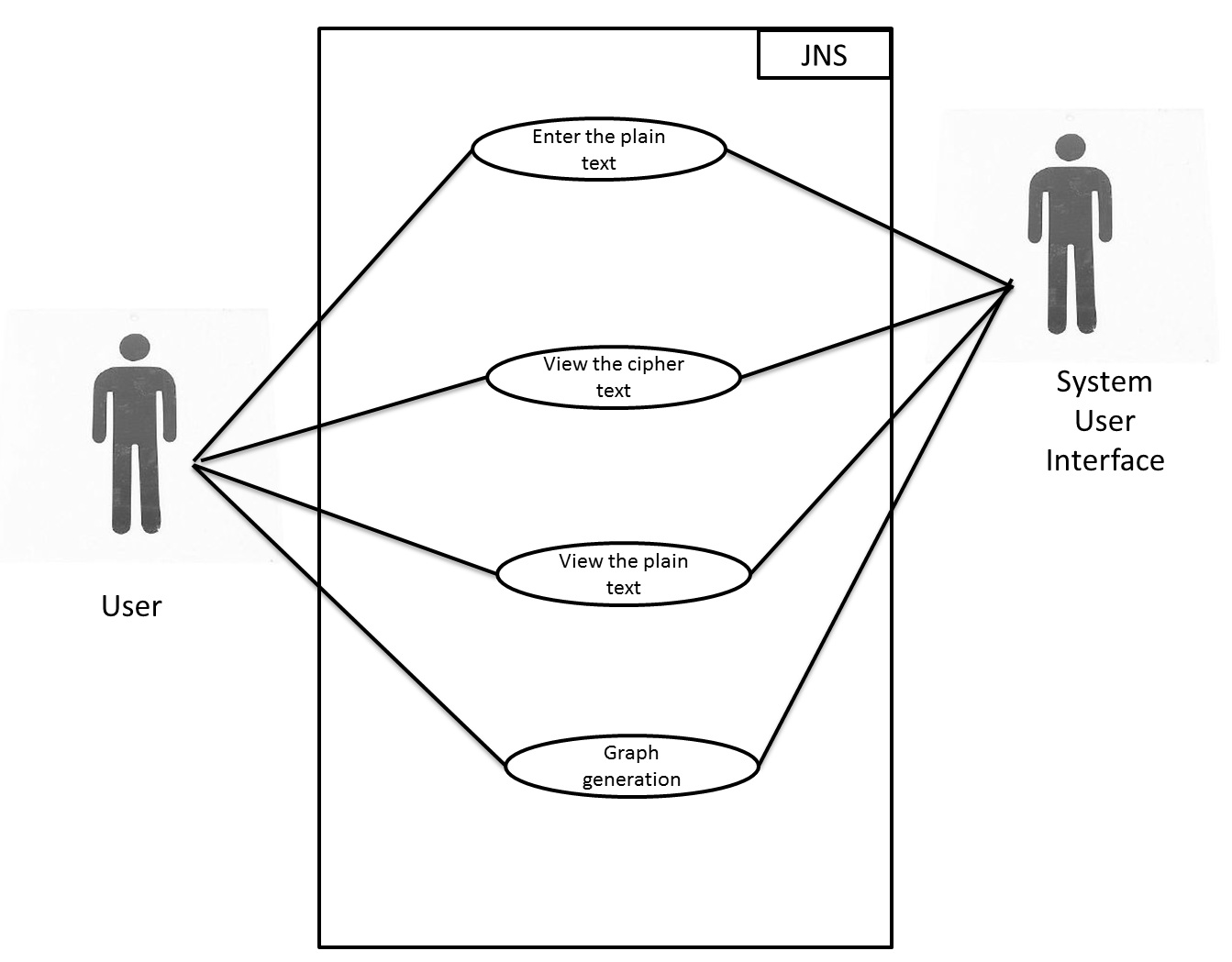
* **Portability**

Since Java is used as the development language, there should be few problems porting the software to new platforms. As an added bonus, as long as the platform contains a java-capable browser, the system will run as an applet.

**4.0 REQUIREMENT GATHERING & ANALYSIS**

**4.1Requirement Elicitation**

**Use Case Diagram and Description**

*****fig 1:Use case diagram*

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **ACTORS** | **USE CASE DESCRIPTION** |
| 1 | User | * Enter the plain text * View the cypher text * View the plain text * Graph generation |
| 2 | System | * Select 4 large prime numbers * Key generation * Storage * Encryption * Decryptions |

*Table . 1: Use Case Description.*

**4.2 FEASIBILITY STUDY**

The very first phase in any system developing life cycle is preliminary investigation. The feasibility study is a major part of this phase. A measure of how beneficial or practical the development of any information system would be to the organization is the feasibility study.

**4.2.1 TECHNICAL FEASIBILITY**

* At least 1.2 GHz Pentium Processor or Intel compatible processor.
* At least 256 MB RAM.
* A mouse or other pointing device.
* At least 10GB free hard disk space.
* At least JAVA Software-1.7
* NetBeans IDE 7.4

**4.2.2ECONOMICAL FEASIBILITY**

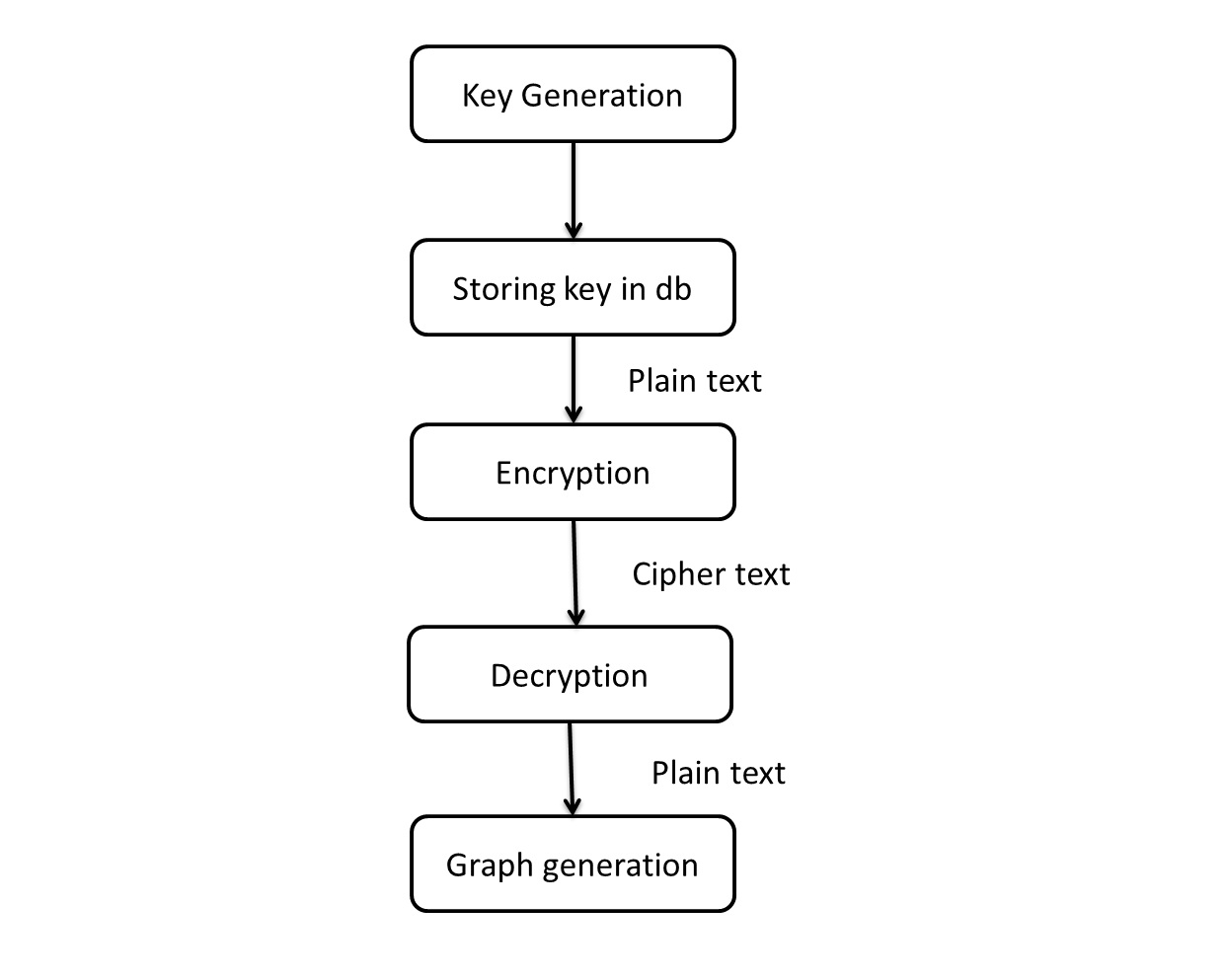
For the user, application will be economically feasible in the following aspects:

Our application will reduce the time that is consumed in the computational operation standard algorithm which will reduce the overall cost.

No additional hardware is required in the proposed system (i.e., it can work on any system which satisfies the minimum requirements of the proposed system).

**4.3 REQUIREMENT ANALYSIS**

**4.3.1 BLOCK DIAGRAM**

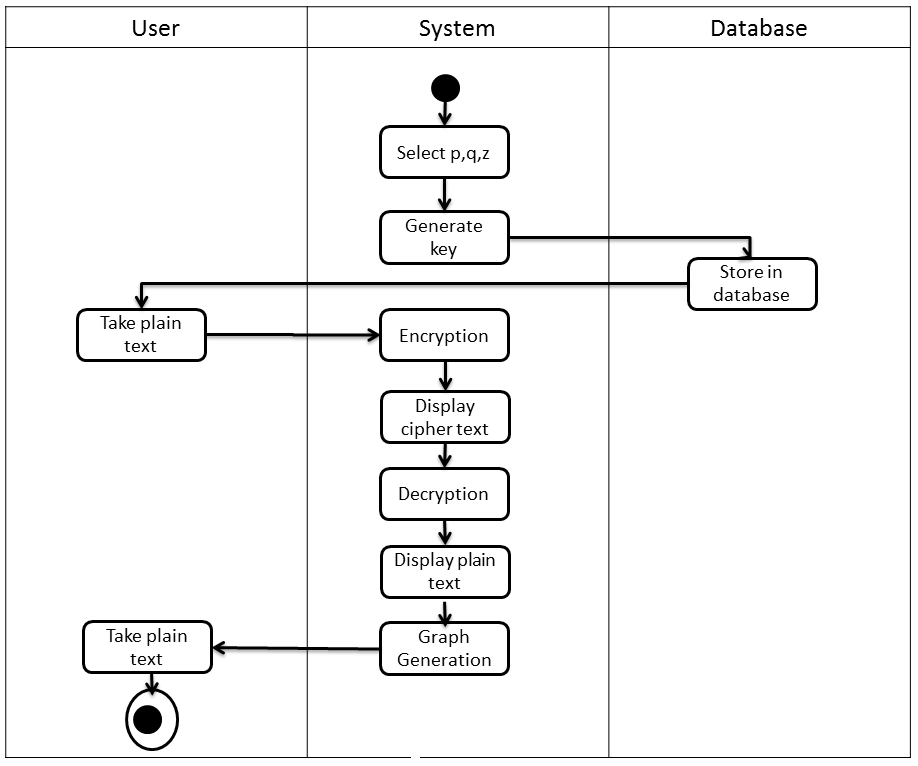


**Fig.2:** *Block Diagram*

**5.0 DESIGN**

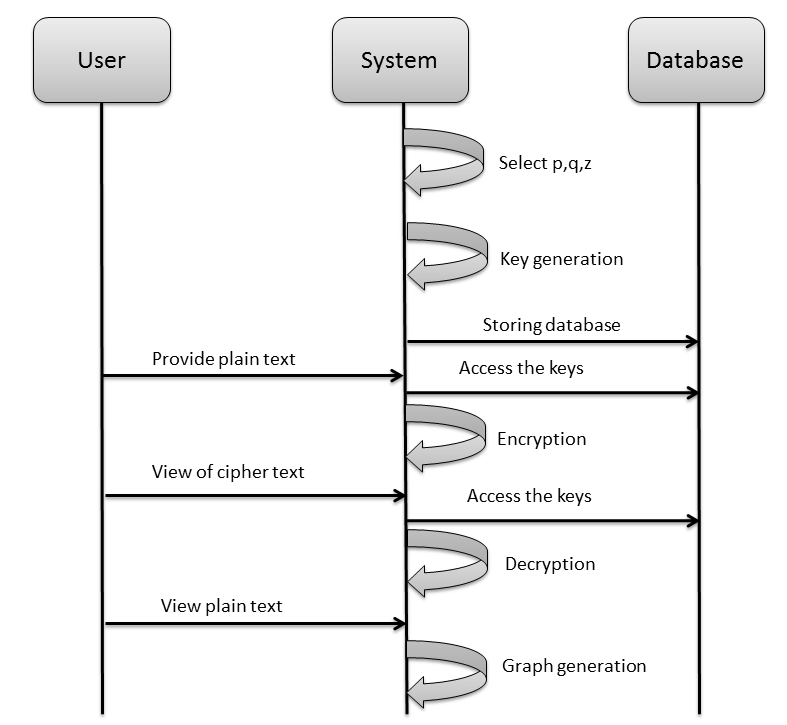
**UML Diagrams**

**Activity Diagram**

****

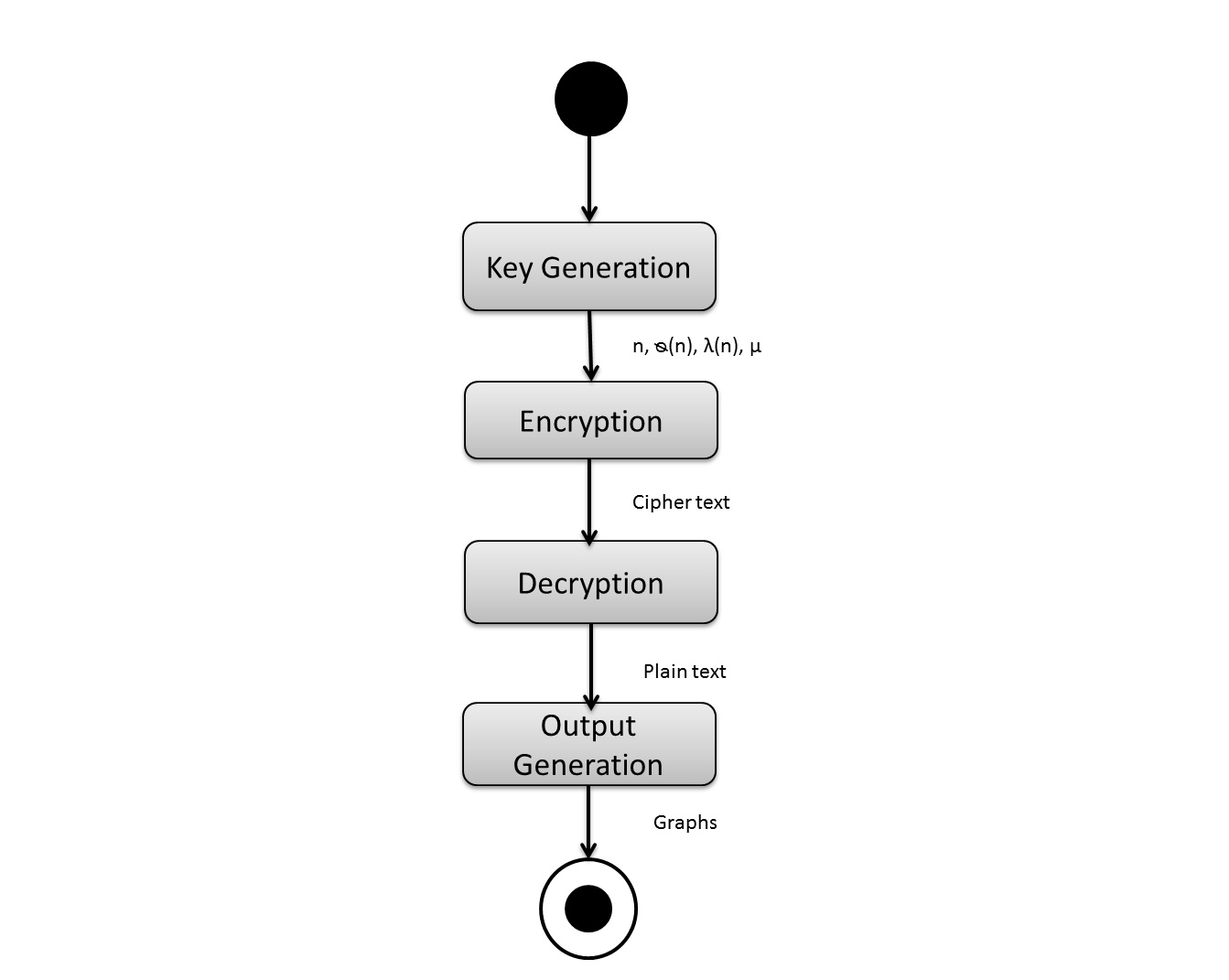
***Fig.3:*** *Activity Diagram*

**Sequence Diagram**

****

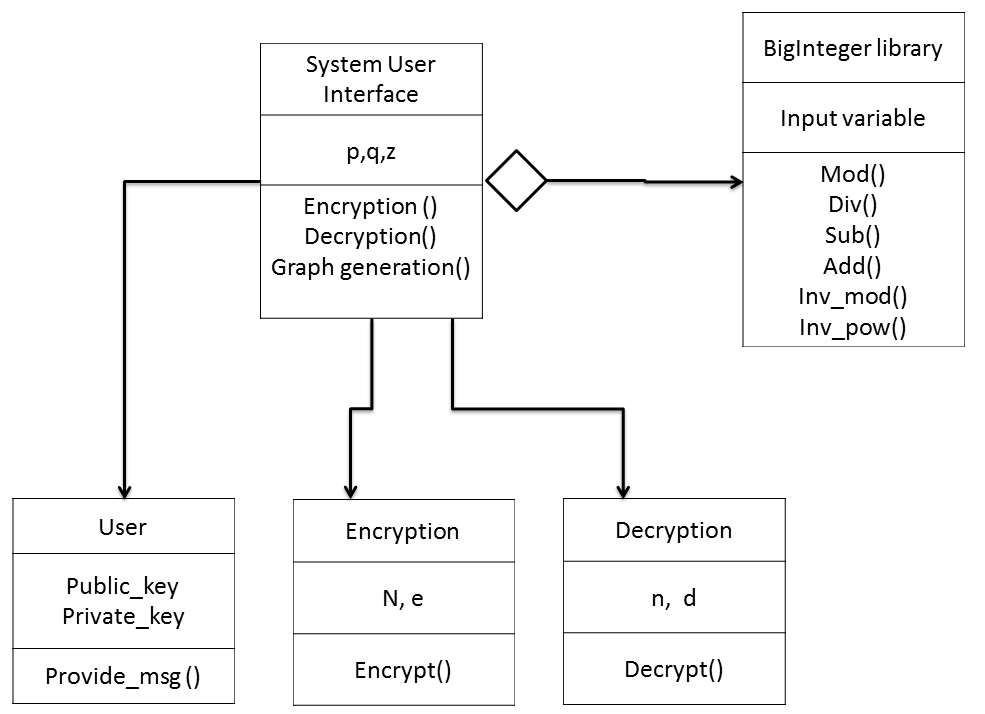
***Fig.4:*** *Sequence Diagram*

**State Chart Diagram**



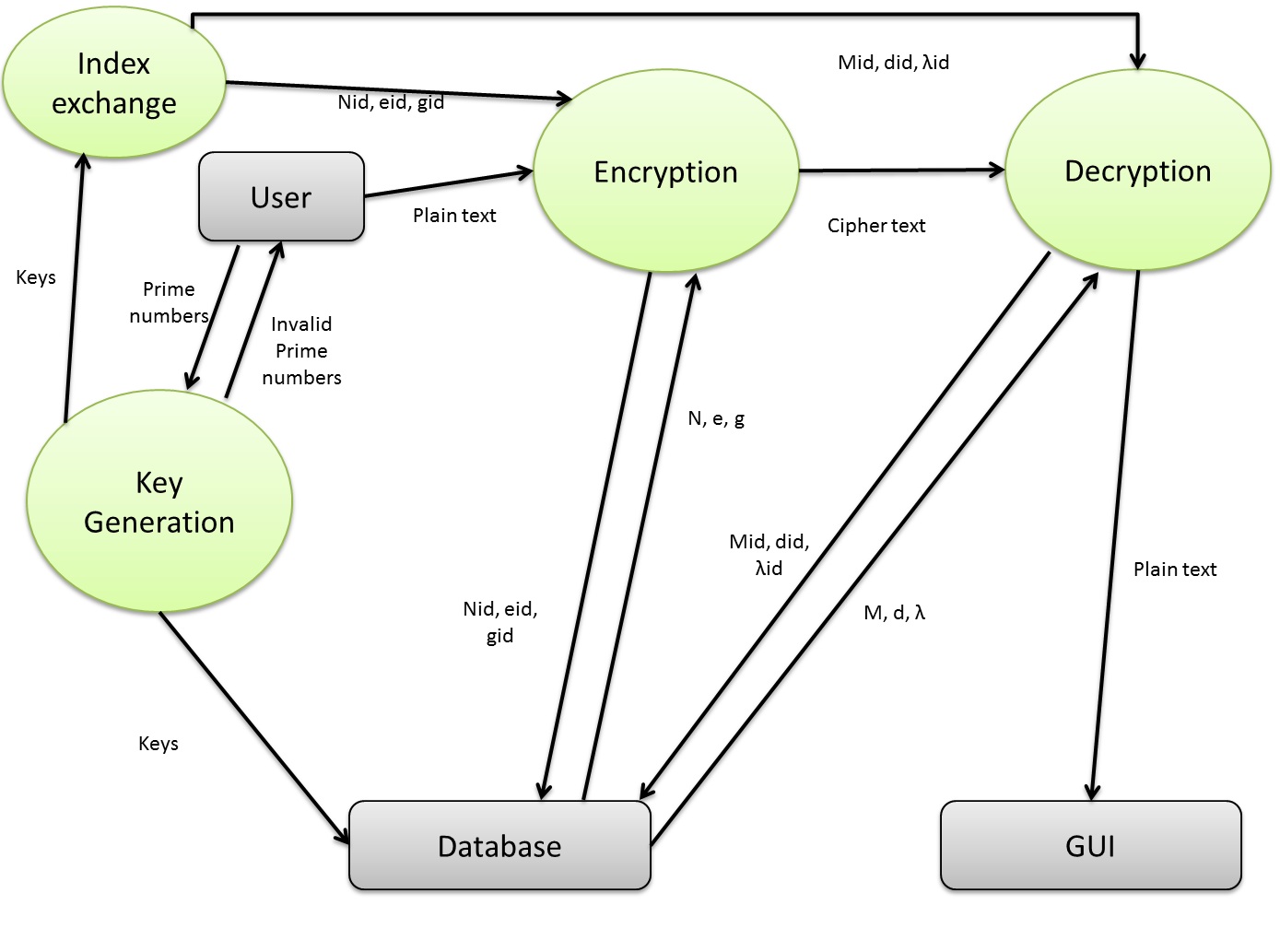
***Fig.5****: State Chart Diagram*

**Class Diagram**

****

***Fig.6****: Class Diagram*

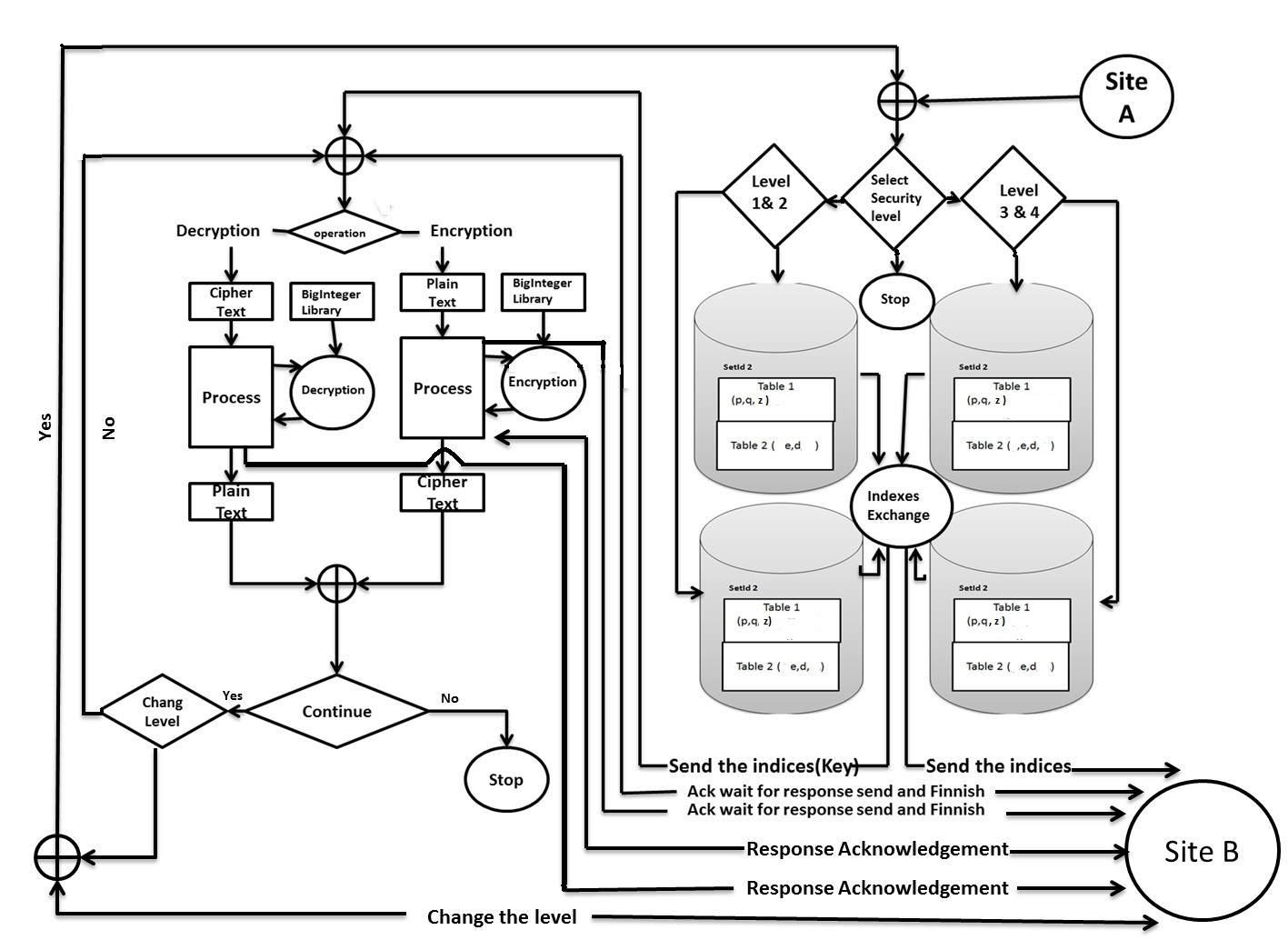
**5.3 Data Flow Diagram**

****

***Fig.7****: Data Flow Diagram*

**6.0 Proposed System**

**6.1 Architecture Diagram**

***Fig.7*** *: Architecture diagram*

**7.0. Implementation**

**7.1 Algorithm**

* Choose three large prime numbers p, q, z randomly and independently of each other. All primes should be of equivalent length.
* By using the database provided the values of n, phi is decided
* They are computed as(in database):
  + Compute n = p x q x z, phi = (p-1) x (q-1) x(z-1)
  + Choose an integer *e*, 1 < e <phi, such that gcd (e,phi) = 1.
  + Compute the secret exponent *d*, 1 < d <phi, such that e x d mod phi=1
* We now exchange Nid , Eid and Did (using RSA Key Pair)
* Retrieve the public key (n, e) from the database.
* Encrypt the data by using the functions of Big Integer library.
* That is **C=Me x mod n**
* Retrieve the public key (d,µ, λ) from the database.
* Decrypt the data by using the functions of big Integer library.
* That is **m=Cd x mod n**

**7.2 TECHNOLOGY USED**

1. **JAVA**



Java is a computer programming language. It enables programmers to write computer instructions using English based commands, instead of having to write in numeric codes. It’s known as a “high-level” language because it can be read and written easily by humans. Like English, Java has a set of rules that determine how the instructions are written. These rules are known as its “syntax”. Once a program has been written, the high-level instructions are translated into numeric codes that computers can understand and execute.

Java was designed with a few key principles in mind:

* Easy to Use: The fundamentals of Java came from a [programming language called c++.](http://cplus.about.com/od/introductiontoprogramming/p/profileofcpp.htm) Although C ++ is a powerful language, it was felt to be too complex in its syntax, and inadequate for all of Java's requirements. Java built on, and improved the ideas of c++, to provide a programming language that was powerful and simple to use.
* Reliability: Java needed to reduce the likelihood of fatal errors from programmer mistakes. With this in mind, [object-oriented programming](http://java.about.com/od/objectorientedprogramming/a/introobjects.htm) was introduced. Once data and its manipulation were packaged together in one place, it increased Java’s robustness.
* Secure: As Java was originally targeting mobile devices that would be exchanging data over networks, it was built to include a high level of security. Java is probably the most secure programming language to date.
* Platform Independent: Programs needed to work regardless of the machine they were being executed on. Java was written to be a portable language that doesn't care about the [operating system](http://cplus.about.com/od/introductiontoprogramming/a/opersys.htm) or the hardware of the computer

1. **MySQL**

MySQL officially is (as of July 2013) the world's second most widely used [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) and most widely used open-source RDBMS. It is named after co-founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, [SQL](http://en.wikipedia.org/wiki/SQL) acronym stands for structured query language.

The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL,[Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_(programming_language))." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL database. include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter,Flickr, and YouTube

1. **NETBEANS**

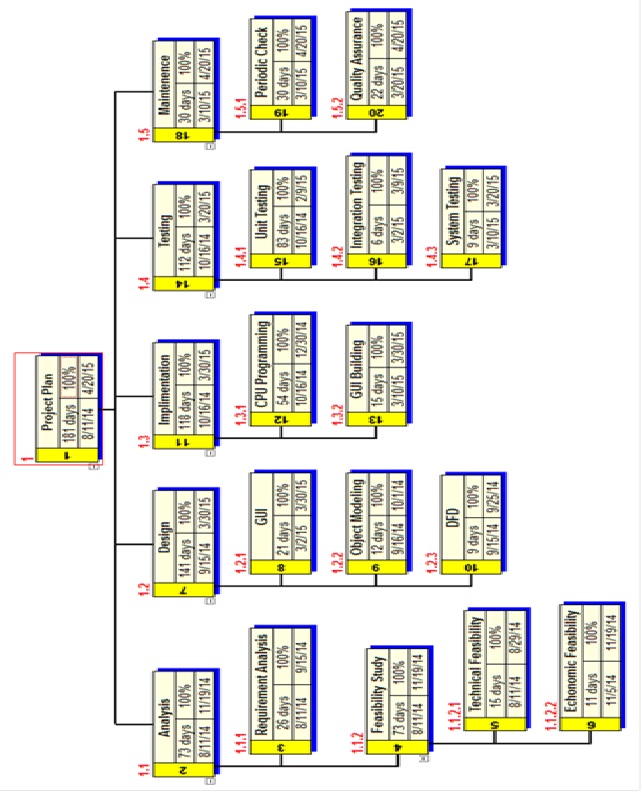
Net Beans  is a [software development](http://en.wikipedia.org/wiki/Software_development) platform written in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)). The NetBeans [Platform](http://en.wikipedia.org/wiki/Platform_(computing)) allows applications to be developed from a set of modular [software components](http://en.wikipedia.org/wiki/Software_component) called *modules*. Applications based on the NetBeans Platform, including the NetBeans [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), can be extended by [third party developers](http://en.wikipedia.org/wiki/Third_party_developer).

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular [PHP](http://en.wikipedia.org/wiki/PHP), [C](http://en.wikipedia.org/wiki/C_(programming_language))/[C++](http://en.wikipedia.org/wiki/C%2B%2B)and [HTML5](http://en.wikipedia.org/wiki/HTML5).

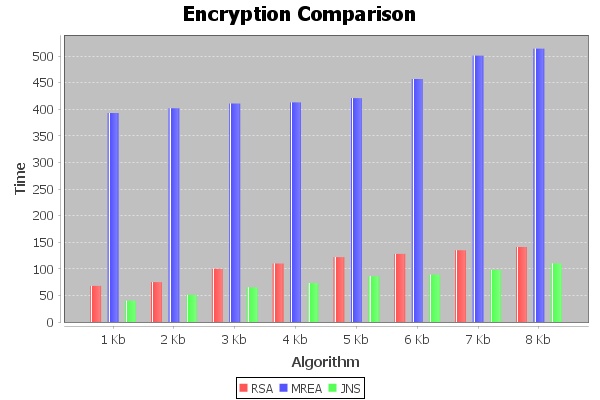
NetBeans is [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) and runs on [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X), [Linux](http://en.wikipedia.org/wiki/Linux), [Solaris](http://en.wikipedia.org/wiki/Solaris_(operating_system)) and other platforms supporting a compatible [JVM](http://en.wikipedia.org/wiki/Java_Virtual_Machine).

The NetBeans Team actively support the product and seek feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback

**7.3 PROJECT WBS CHART**

*Fig no 8: Time Line Chart*

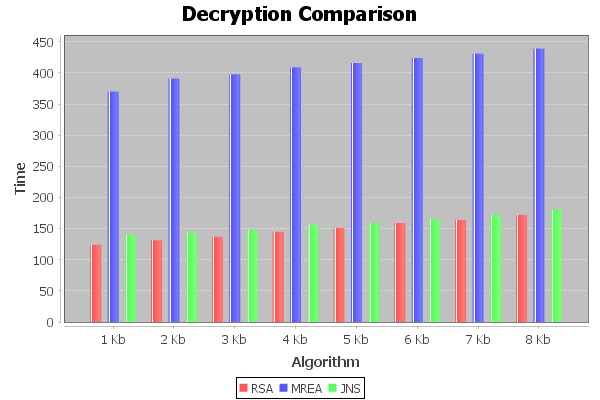
**8.0 Results and Discussions**



*Fig no 9 : Static Encryption Comparison Graph*

**Encryption Comparison :**

The Graph gives a very clear idea about the time complexity for encryption of the three algorithm i.e RSA , MREA and JNS which are represented using red ,blue and green colour respectively. It’s evident from the graph the difference of the time complexity of these algorithm. JNS algorithm drastically decreases the time complexity which is otherwise taken by MREA thereby making it practically usable.



*Fig no 10 : Static Decryption Comparison Graph*

**Decryption Comparison:**

The above shown graph gives the idea about the decryption time of each of the three algorithms. Due to the usage of offline storage and BigInteger library for JNS algorithm the time complexity of it decreased to less than half of that taken by the MREA algorithm. But we see that there’s no much difference between the decryption time of RSA and JNS but the security of the latter is than RSA.

.

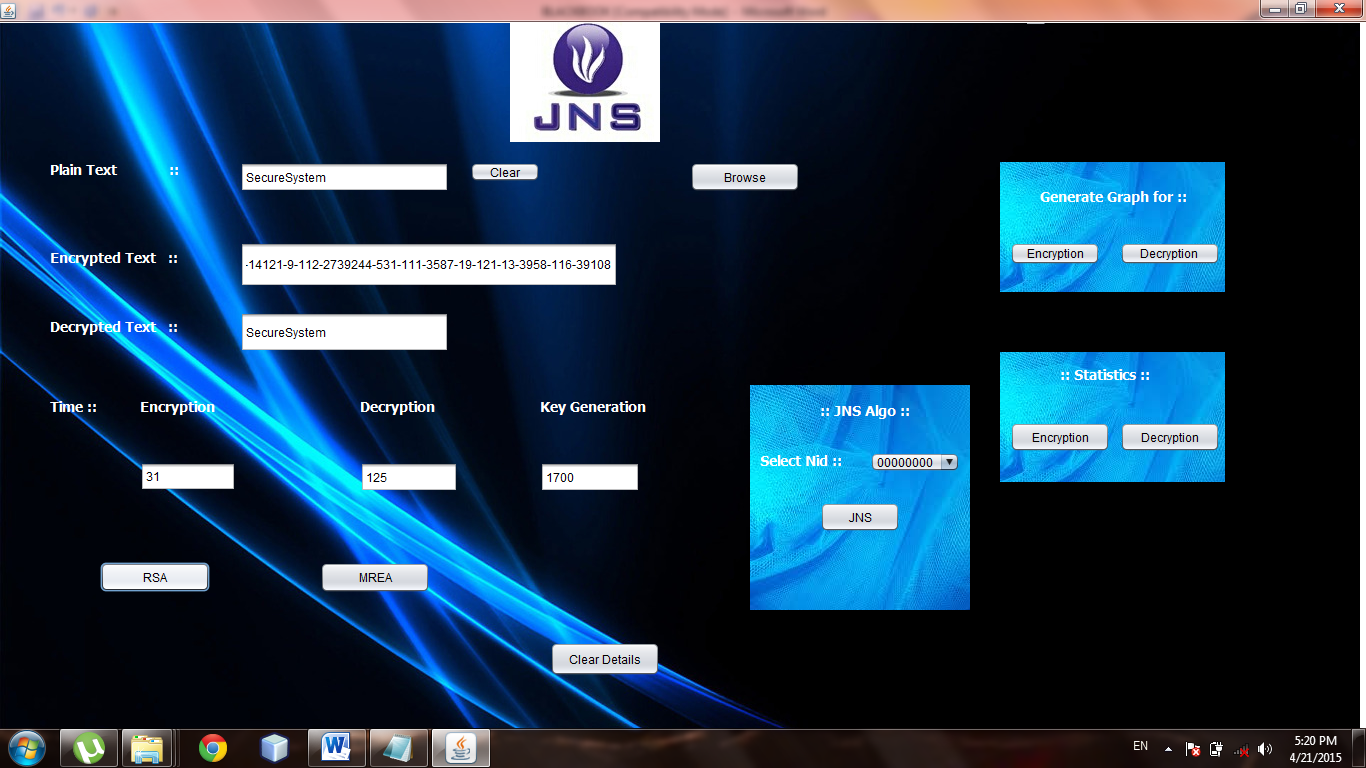
* The above shown graph gives a very clear idea that the JNS algorithm provides an intermediate level of security and time complexity as compared to RSA and MREA
* RSA takes less time to encrypt and decrypt data but according to the current statistics the security could be breached
* Whereas MREA provides a high level of security but when compared to the time complexity the security it provides is negligible
* So we can thus conclude that JNS is a better algorithm when time and security both can’t be compromised with and the application requires harmony between the above two parameters.
* Also the JNS requires three prime numbers for key generation as compared to two and four for RSA and MREA
* When hacking an Encrypting algorithm the hacker tries to find the value of n. Once he becomes successful in doing so then factors of n is found and through a series of brute force attack the hacking is done
* In other words the probability of finding the factors is where the strength of the algorithm lies.
* RSA uses two prime numbers so if one prime number is decided the other can be easily achieved by dividing the first value with n.
* Whereas JNS uses three and thus it becomes difficult to hack into the system since the probability of finding two factors is more than three.

**8.1 Home Page**

****

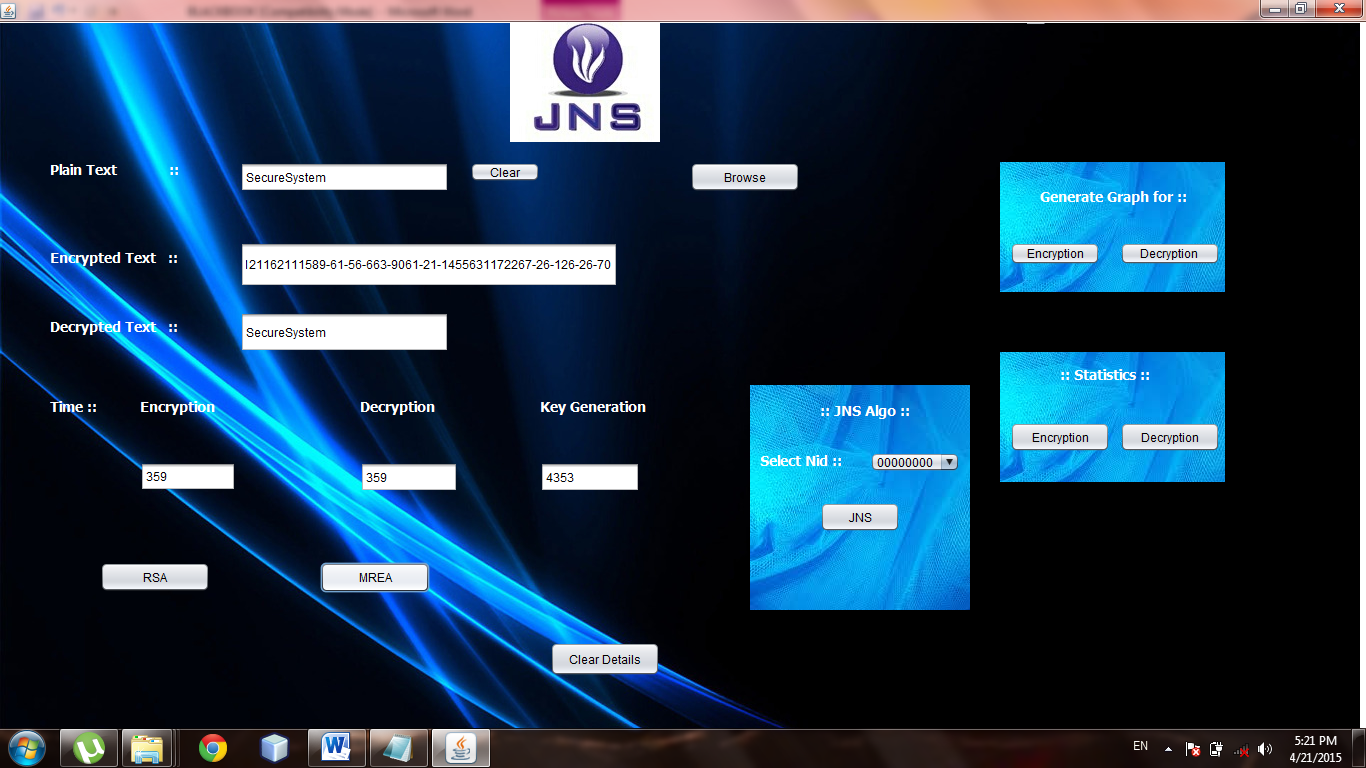
*Fig no 12: Home Page*

**8.2 RSA Output**



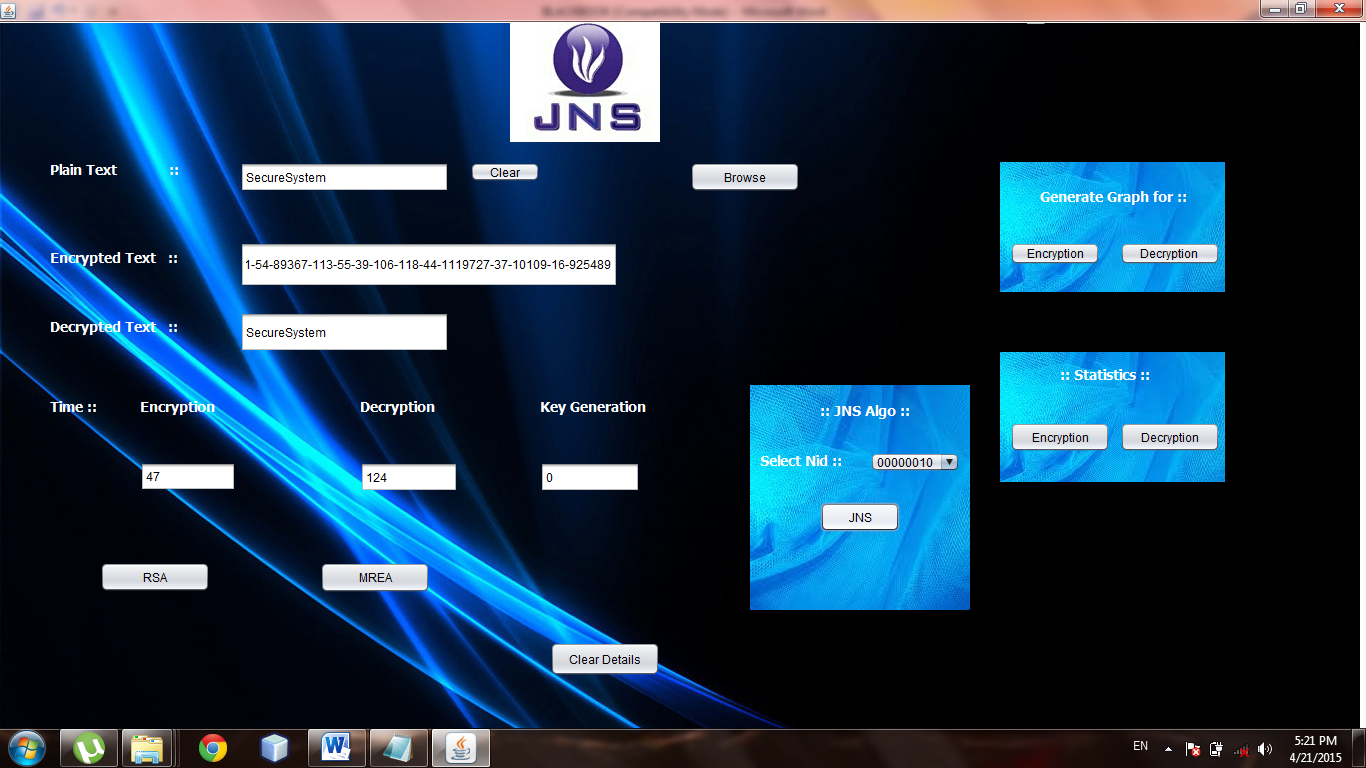
*Fig no 13 : RSA Output.*

**8.3 MREA Output**



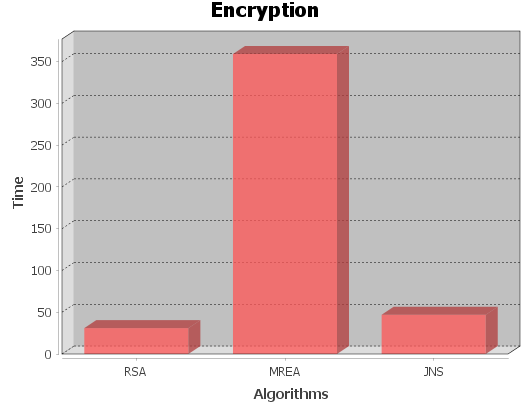
*Fig no14 : MREA Output*

**8.4 JNS Output**



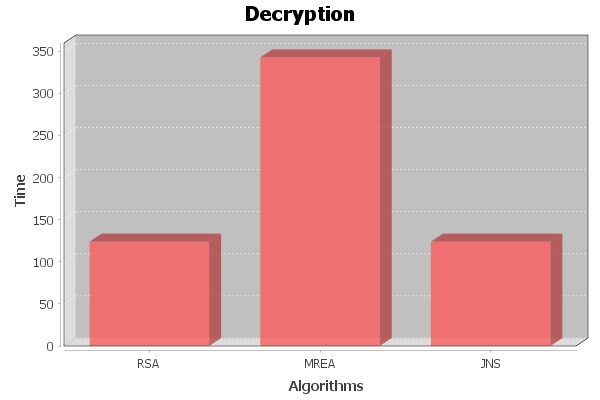
*Fig no15 : JNS Output.*

**8.5 Encryption Graph**

****

*Fig no 16: Dynamic Encryption Comparison Graph.*

**8.6 Decryption Graph**

****

*Fig no 17: Dynamic Decryption Comparison Graph*

*.*

**9.0 Conclusion and Further Work**

**9.1 Conclusion**

The aim of JNS was to devise an algorithm which is efficient when compared on the two main parameters of security and time complexity. According to the statistics the RSA security system can be breached by 2020

And the time taken by MREA algorithm makes all the application based on MREA impossible

Whereas JNS takes more time than RSA but the time is not much as compared to the security it provides.

Though JNS is comparatively less secure than MREA the time complexity of both algorithm gives us the idea that MREA can have no practical application whereas JNS can be used for practical application

**9.2 Further Work**

* JNS is a powerful algorithm that provides an intermediate level of security and time complexity as compared to RSA and MREA.
* JNS algorithm can further be modified by using recent and efficient technology like Membrane Computing
* The “Indexes Exchange” as proposed can also get better result.
* JNS algorithm can prove to be as revolutionary technique for practical applications.

**10.0. References**

1. Ravi Shankar Dhakarand and PrashantSharma, “modified RSA encryption algorithm (MREA)”. Amit Kumar Gupta Sbcet, Jaipur, Rajasthan, India, 2012.
2. RajorshiBiswasShibdas and BandyopadhyayAnirban Banerjee, “A Fast Implementation Of
3. The RSA Algorithm Using the GNU MP Library”, iiit-Calcutta, 2010.
4. MS. RituPatidar and Mrs.RupaliBhartiya, “Offline Storage and Prime Number “, Institute of
5. Technology & Science, Indore, India
6. Salah Zaher, amrbadr& Ibrahim Farag, “Performance Enhancement Of RSA Cryptography
7. Algorithm By Membrane Computing”*, Cario Egypt.*
8. EvgenyMilanov,” The RSA Algorithm”, 3 June 2009.

*William Stallings,”Cryptology and Network Security”, an imprint of Pearson Education*

**11.0. Appendix**

|  |  |
| --- | --- |
| **Abbreviations** | **Definitions** |
|
| **RSA** | Cryptic algorithm |
| **MREA** | Modified RSA algorithm |
| **JNS** | Proposed algorithm |

*Table 2 : Abbreviations*

* 1. Cryptography – Cryptography (or cryptology) is the practice and study of techniques for [secure communication](http://en.wikipedia.org/wiki/Secure_communication) in the presence of third parties (called [adversaries](http://en.wikipedia.org/wiki/Adversary_%28cryptography%29))
  2. Encryption - In [cryptography](http://en.wikipedia.org/wiki/Cryptography), encryption is the process of encoding messages or information in such a way that only authorized parties can read it.
  3. Cipher - In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure.
  4. Plaintext - In [cryptography](http://en.wikipedia.org/wiki/Cryptography), plaintext is information a sender wishes to transmit to a receiver. Clear text is often used as synonym. Plaintext has reference to the operation of [cryptographic](http://en.wikipedia.org/wiki/Cryptography) algorithms, usually [encryption](http://en.wikipedia.org/wiki/Encryption) algorithms, and is the input upon which they operate. Cleartext, by contrast, refers to data that is transmitted or stored unencrypted (that is, 'in the clear').
  5. Cipher text - In [cryptography](http://en.wikipedia.org/wiki/Cryptography), cipher text (or cypher text) is the result of [encryption](http://en.wikipedia.org/wiki/Encryption) performed on [plaintext](http://en.wikipedia.org/wiki/Plaintext) using an algorithm, called a [cipher](http://en.wikipedia.org/wiki/Cipher).
  6. Cryptographic Key - A cryptographic key is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa. This key remains private and ensures secure communication.