****

**Hybrid Approach for security enhancement using irrational number**

Mr. Monil Parikh

Mr. Yogesh Patel

Mr. Chinar Vartak

UNIVERSITY OF MUMBAI

THAKUR COLLEGE OF ENGINEERING AND TECHNOLOGY

December ‘11

***CERTIFICATE***

*This is to certify that the report titled:*

“**Hybrid Approach for security enhancement using irrational number** ”

*Submitted by*

Mr. Monil Parikh

Mr. Yogesh Patel

Mr. Chinar Vartak

*embodies the bonafide work done at*

TCET, Mumbai

*by them under my guidance and supervision.*

*Signature of the Internal Guide*

*Name:* ………………………..

*Designation: ………………….*

*Date: ………………………….*

*This is to certify that*

Mr. Monil Parikh

Mr. Yogesh Patel

Mr. Chinar Vartak

*have satisfactorily completed the requirements of the*

**Term Work OF PROJECT**

*as prescribed by the*

**University of Mumbai.**

Project Coordinator. HOD IT Principal

*(Mr. Vinayak Bharadi) (Mrs. Kamal Shah) (Dr. B. K. Mishra)*

*This report has been examined by us as per the University requirements at*

**TCET, Mumbai** *on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Signature……………………… Signature…………………….*

*Name………………………… Name…………………………*

**Internal Examiner External Examiner**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTENTS** | | | |
| Sr. No | Sub. Sec. | **TOPIC** | Pg No |
| 1. |  | Acknowledgement | 4 |
| 2. |  | Abstract | 5 |
| 3. |  | Overview | 7 |
| 4 |  | Module Description | 12 |
| 5. |  | Planning | 14 |
| 6. |  | Gantt Chart | 16 |
| 7. |  | Formulation | 17 |
|  | 7.1 | Use Case Diagrams | 17 |
|  | 7.2 | Class Diagram | 18 |
|  | 7.3 | Sequence Diagrams | 19 |
|  | 7.4 | Data Flow Diagrams | 20 |
| 8 |  | Progress/Screenshots | 24 |
| 9 |  | Current Progress | 26 |
| 10 |  | Conclusion | 28 |
|  | 10.1 | Benefits | 28 |
|  | 10.2 | Features | 28 |
|  |  | References | 29 |

**1. ACKNOWLEDGEMENT**

We take this opportunity to thank Prof. Vikas Kaul,our internal guide for his constant support, co-ordination and encouragement. We also thank him for his moral support and patience during the course of the project. His expertise in the subject of Network Security was a great boost to our efforts. We extend our sincere thanks to our Head of Department, Dr. Kamal Shah.

We would also like to thank our principal Dr. B.K. Mishra , our project co-ordinators and all the staff members who helped us in our project.

We are also grateful to our friends for their unconditional support and sincere feedback about our project.

Our sincere thanks to the management of THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY for giving us required facilities to accomplish our project work.

**NAMES OF STUDENTS:**

Monil Parikh

Yogesh Patel

Chinar Vartak

**2. ABSTRACT**

DES (Data Encryption Standard) is a cryptographic standard. However, the applications of it are limited because of the small key space. Based on irrational numbers, an improved scheme that enhances the randomness of sub-Key is proposed, in which the permutation is controlled by irrational number which is considered as false chaos. Moreover, the permutation controlled by data can be performed at high speed in generic CPU. It is shown that this scheme can expand the key space without costing any more time to run.

The plaintext is encrypted to cipher text by the key with a length of 64 bits, in which 56 bits are used for encryption, and the others are employed for parity test. Encryption and decryption use the same algorithm as well as the key. It is no longer a question to attack the 56-bit key with the development of computer technology. The attacker can decipher DES within 20 hours through exhaustive key search.

An improved scheme is proposed here to control the transposition by irrational number in order to overcome the defects mentioned above. Irrational numbers, which are similar to data sequences generated form chaotic system, are infinite and non-cycle. However the former is not sensitive to the initial condition. Therefore, using irrational numbers to encrypt will provide higher security without strict chaos synchronization.

The improved scheme prevents the key from being directly involved in the production of sub-keys, and increases the transposition controlled by irrational number on the basis of DES, which has higher security, great non-crack ability and wide applicability.

To further improve non-crack ability we integrate AES(Advanced Encryption Standard) with Improved DES with Irrational Numbers. We will integrate AES in between all rounds of DES encryption. The result of EX-OR of fiestal function and left 32-bit part of 64-bit data will be encrypted with AES and left 32-bit of original 64-bit data will also be encrypted with AES. These two AES encrypted 32-bit blocks will be used in further rounds.

**3. OVERVIEW**

DES (Data Encryption Standard) is a cryptographicstandard, which was collected publicly by ANSI, recommended by IBM and used for data encryption. It became America data encryption standard with the approval of ANSI in 1997.7. The plaintext is encrypted to cipher text by the key with a length of 64 bits, in which 56 bits are used for encryption, and the others are employed for parity test. Encryption and decryption use the same algorithm as well

as the key.

**HISTORY OF CRYPTOGRAPHY**:

The history of cryptography begins thousands of years ago. Until recent decades, it has been the story of what might be called classic cryptography — that is, of methods of encryption that use pen and paper, or perhaps simple mechanical aids. In the early 20th century, the invention of complex mechanical and electromechanical machines, such as the Enigma rotor machine, provided more sophisticated and efficient means of encryption; and the subsequent introduction of electronics and computing has allowed elaborate schemes of still greater complexity, most of which are entirely unsuited to pen and paper.

The development of cryptography has been paralleled by the development of cryptanalysis — the "breaking" of codes and ciphers. The discovery and application, early on, of frequency analysis to the reading of encrypted communications has, on occasion, altered the course of history.

Until the 1970s, secure cryptography was largely the preserve of governments. Two events have since brought it squarely into the public domain: the creation of a public encryption standard (DES), and the invention of public-key cryptography.

**DES:**

The Data Encryption Standard (DES) is a block cipher that uses shared secret encryption. It was selected by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976 and which has subsequently enjoyed widespread use internationally. It is based on a symmetric-key algorithm that uses a 56-bit key. The algorithm was initially controversial because of classified design elements, a relatively short key length, and suspicions about a National Security Agency (NSA) backdoor. DES consequently came under intense academic scrutiny which motivated the modern understanding of block ciphers and their cryptanalysis.

DES is the archetypal block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is never quoted as such. Every 8th bit of the selected key is discarded, that is, positions 8, 16, 24, 32, 40, 48, 56, 64 are removed from the 64 bit key leaving behind only the 56 bit key.

The algorithm's overall structure is shown in Figure 1: there are 16 identical stages of processing, termed rounds. There is also an initial and final permutation, termed IP and FP, which are inverses (IP "undoes" the action of FP, and vice versa). IP and FP have almost no cryptographic significance, but were apparently included in order to facilitate loading blocks in and out of mid-1970s hardware.

Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme. The Feistel structure ensures that decryption and encryption are very similar processes — the only difference is that the subkeys are applied in the reverse order when decrypting. The rest of the algorithm is identical. This greatly simplifies implementation, particularly in hardware, as there is no need for separate encryption and decryption algorithms.

The ⊕ symbol denotes the exclusive-OR (XOR) operation. The F-function scrambles half a block together with some of the key. The output from the F-function is then combined with the other half of the block, and the halves are swapped before the next round. After the final round, the halves are not swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.

The Feistel (F) function:

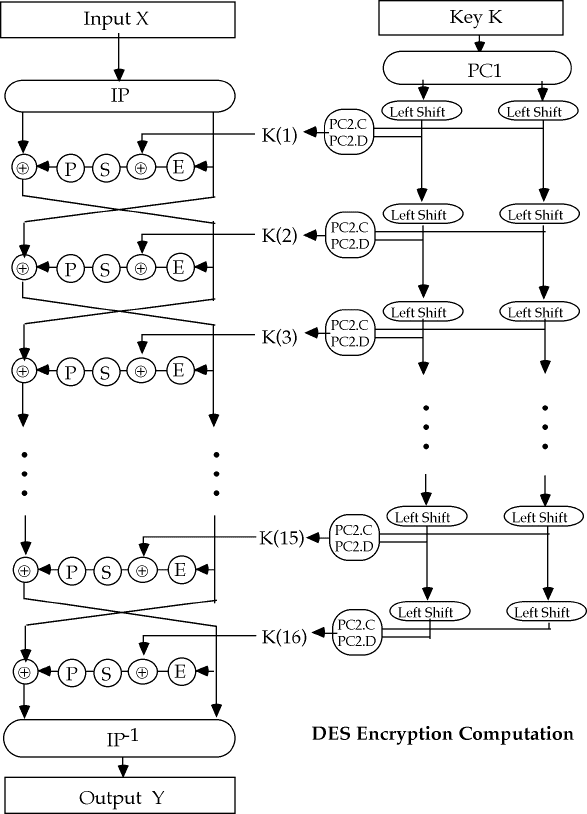
The F-function, depicted in Figure 2, operates on half a block (32 bits) at a time and consists of four stages:

1. Expansion — the 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit(8\*6=48bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

2. Key mixing — the result is combined with a subkey using an XOR operation. 16 48-bit subkeys — one for each round — are derived from the main key using the key schedule (described below).

3. Substitution — after mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES — without them, the cipher would be linear, and trivially breakable.

4. Permutation — finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. This is designed so that, after expansion, each S-box's output bits are spread across 6 different S boxes in the next round.

****

**AES :**

AES is based on a design principle known as a [Substitution permutation network](http://en.wikipedia.org/wiki/Substitution-permutation_network). It is fast in both [software](http://en.wikipedia.org/wiki/Computer_software) and [hardware](http://en.wikipedia.org/wiki/Hardware).[[6]](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-5) Unlike its predecessor, DES, AES does not use a [Feistel network](http://en.wikipedia.org/wiki/Feistel_network).

AES has a fixed [block size](http://en.wikipedia.org/wiki/Block_size_%28cryptography%29) of 128 [bits](http://en.wikipedia.org/wiki/Bit) and a [key size](http://en.wikipedia.org/wiki/Key_size) of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The blocksize has a maximum of 256 bits, but the keysize has no theoretical maximum.

AES operates on a 4×4 [column-major order](http://en.wikipedia.org/wiki/Column-major_order) matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special [finite field](http://en.wikipedia.org/wiki/Finite_field_arithmetic).

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

### High-level description of the algorithm

1. KeyExpansion—round keys are derived from the cipher key using [Rijndael's key schedule](http://en.wikipedia.org/wiki/Rijndael_key_schedule)
2. Initial Round
   1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor
3. Rounds
   1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a [lookup table](http://en.wikipedia.org/wiki/Rijndael_S-box).
   2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
   3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
   4. AddRoundKey
4. Final Round (no MixColumns)
   1. SubBytes
   2. ShiftRows
   3. AddRoundKey

### The SubBytes step

In the SubBytes step, each byte in the matrix is updated using an 8-bit [substitution box](http://en.wikipedia.org/wiki/Substitution_box), the [Rijndael S-box](http://en.wikipedia.org/wiki/Rijndael_S-box). This operation provides the non-linearity in the [cipher](http://en.wikipedia.org/wiki/Cipher). The S-box used is derived from the [multiplicative inverse](http://en.wikipedia.org/wiki/Multiplicative_inverse) over [**GF**](http://en.wikipedia.org/wiki/Finite_field)(*28*), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible [affine transformation](http://en.wikipedia.org/wiki/Affine_transformation). The S-box is also chosen to avoid any fixed points (and so is a [derangement](http://en.wikipedia.org/wiki/Derangement)), and also any opposite fixed points.

### The ShiftRows step

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain [offset](http://en.wikipedia.org/wiki/Offset_%28computer_science%29). For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

### The MixColumns step

In the MixColumns step, the four bytes of each column of the state are combined using an invertible [linear transformation](http://en.wikipedia.org/wiki/Linear_transformation). The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides [diffusion](http://en.wikipedia.org/wiki/Diffusion_%28cryptography%29) in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

\begin{bmatrix}
2 & 3 & 1 & 1 \\
1 & 2 & 3 & 1 \\
1 & 1 & 2 & 3 \\
3 & 1 & 1 & 2
\end{bmatrix}.

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing [xor](http://en.wikipedia.org/wiki/Xor) with the initial unshifted value. After shifting, a conditional [xor](http://en.wikipedia.org/wiki/Xor) with 0x11B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over **GF**(*28*) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03 · x3 + x2 + x + 0x02. The coefficients are displayed in their [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal) equivalent of the binary representation of bit polynomials from **GF**(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular [MDS matrix](http://en.wikipedia.org/wiki/MDS_matrix) in a [finite field](http://en.wikipedia.org/wiki/Finite_field). This process is described further in the article [Rijndael mix columns](http://en.wikipedia.org/wiki/Rijndael_mix_columns).

### The AddRoundKey step

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main [key](http://en.wikipedia.org/wiki/Key_%28cryptography%29) using [Rijndael's key schedule](http://en.wikipedia.org/wiki/Rijndael_key_schedule); each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise [XOR](http://en.wikipedia.org/wiki/Exclusive_or).



**MATLAB:**

MATLAB (matrix laboratory) is a [numerical computing](http://en.wikipedia.org/wiki/Numerical_analysis) environment and [fourth-generation programming language](http://en.wikipedia.org/wiki/Fourth-generation_programming_language). Developed by [MathWorks](http://en.wikipedia.org/wiki/MathWorks), MATLAB allows [matrix](http://en.wikipedia.org/wiki/Matrix_%28mathematics%29) manipulations, plotting of [functions](http://en.wikipedia.org/wiki/Function_%28mathematics%29) and data, implementation of [algorithms](http://en.wikipedia.org/wiki/Algorithm), creation of [user interfaces](http://en.wikipedia.org/wiki/User_interface), and interfacing with programs written in other languages, including [C](http://en.wikipedia.org/wiki/C_%28programming_language%29), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29), and [Fortran](http://en.wikipedia.org/wiki/Fortran).

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications. Features include:

* High-level language for technical computing
* Development environment for managing code, files, and data
* Interactive tools for iterative exploration, design, and problem solving
* Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
* 2-D and 3-D graphics functions for visualizing data
* Tools for building custom graphical user interfaces
* Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft® Excel®

MATLAB is a [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) product of MathWorks, so users are subject to [vendor lock-in](http://en.wikipedia.org/wiki/Vendor_lock-in).Although MATLAB Builder can deploy MATLAB functions as library files which can be used with [.NET](http://en.wikipedia.org/wiki/.NET_Framework) or [Java](http://en.wikipedia.org/wiki/Java_%28software_platform%29) application building environment, future development will still be tied to the MATLAB language.

**LITERATURE SURVEY:**

Technical paper:

Referencing IEEE Int. Conference paper proposed by *Jing Wang*, *Guo-ping Jiang* helped us to motivate to undertake the project. Here based on irrational numbers, an improved scheme that enhances the randomness of sub-Key is proposed, in which the permutation is controlled by irrational number which is considered as false chaos.

Referencing IEEE Int. Conference paper proposed by Eashwar Thiagarajan and Madhuri Gourishetty helped us to motivate to undertake the project.

**BOOKS:**

# [9] Forouzan Behrouz for Cryptography & Network Security (McGraw-Hill Forouzan Networking) volume 2 and Network Security: The Complete Reference

Mark Rhodes-Ousley (Author), Roberta Bragg (Author), Keith Strassberg (Author)

# helped us to understand the working of DES algorithm in its purest way possible

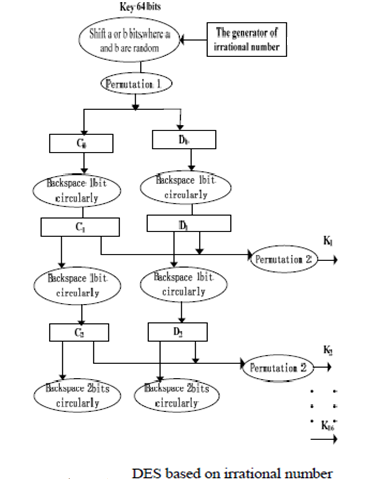
**4. MODULE DESCRIPTION**

The 64-bit key is controlled by irrational number to shift before the sub-key being produced. Therefore the key is not directly involved in the production of the sub-keys. In order to increase the randomness of sub-key, the production and the selection of ‘a’ and ‘b’ are all controlled by irrational number. The Location information of ‘a’ and ‘b’, used for shifting, is transmitted to the receiver through the **safe channel**. The location information will be one part of the key.

In order to increase the randomness of sub-key, the production and the selection of ‘a’ and ‘b’ are all controlled by irrational number. The Location information of ‘a’ and ‘b’, used for shifting, is transmitted to the receiver through the safe channel. The location information will be one part of the key.For example, selecting two 2-digit decimal numbers after the decimal point of the irrational number π randomly, then let ‘a’ be equal to the first number, and ‘b’ be equal to the other number. Another two number which 629 are selected in the same way are made XOR (exclusive-or). If the result is odd, there will be ‘a’ bits shifted, otherwise, ‘b’ bits will be shifted.

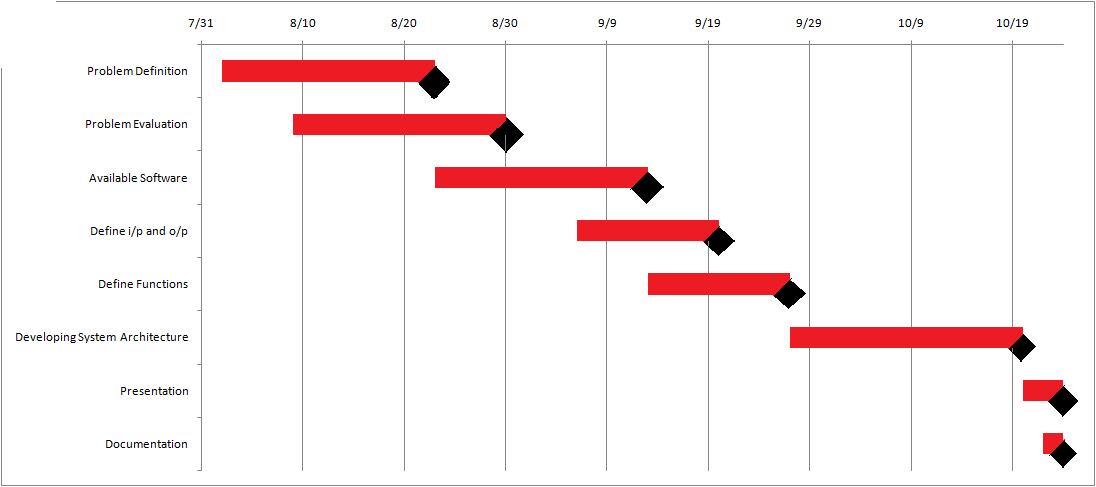
Firstly, the running time that DES took as much as that of the DES based on irrational number, i.e., the **confidentiality of the key is enhanced** without spending more time. Secondly, based on the same plain text and key , the cipher text of DES after several simulations is the same, but it is random about the DES controlled by irrational number, i.e., the **key space is expanded** through increasing the randomness of sub-keys. Finally, we can draw the conclusion that the algorithm will have wider foreground applications based on these advantages including the higher security, great non-crack ability and wide applicability.

Thus, DES is a common encryption method. Due to the smaller key space, the applications of DES are limited. It is our purpose that designs an encryption algorithm with better security, bigger secret-key space and higher encrypting efficiency. An improved scheme based on irrational number is proposed in our final year project. Simulation results show that this scheme can expand the key space without any more running time and enhance the security of the encryption algorithm.



**5. PLANNING**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SN** | **Task** | **Subtask** | **Activity** | **Duration (hrs)** | **Start Date** | **End Date** |
| 1.  2.  3.  4.  5.  6.  7.  8. | Problem  Definition.  Problem evaluation.  Research availability of available software.  Define input and desired output.  Define functions & Behaviour.  Development  of system architecture.  Presentation  Creation of  Documentation | Formulation  of the process statement.  Searching for multiple alternative solutions of main objective.  Research the components available.  Defining the attributes and data.  Describe modes of interaction.  Describe interface.  Planning the  logical execution of the system being developed and analyzing software and hardware requirements.  Giving  a seminar about current status of work done to internal guide, teachers and  also the HOD.  Completely analyzing the work done till date | Brain storming session amongst  the group members.  Discussion and searching on internet.  Identify the reusable components.  Describe the Input data required and output according to the software.  Analyzing the concept in terms of functions.  Formulated the general idea of the working of the process. Visualized a Standard working solution which satisfies our goals and objectives.  Creation of power point presentation  on Microsoft power point.  Formatting the documentation to a desirable need. | 5 hrs    10 hrs  5 hrs  5 hrs    10 hrs      20 hrs  5 hrs  10 hrs | 02-8-2011  09-8-2011  2-09-2011  09-09-2011  13-09-2011  27-09-2011  14-10-2011  21-10-2010 | 23-08-2010  11-08-2011  2-09-2011  20-09-2011  27-09-2011  12-10-2011  14-10-2011  22-10-2010 |

 **6. GANTT CHART**

**7. FORMULATION**

7.1 Use Case Diagrams:

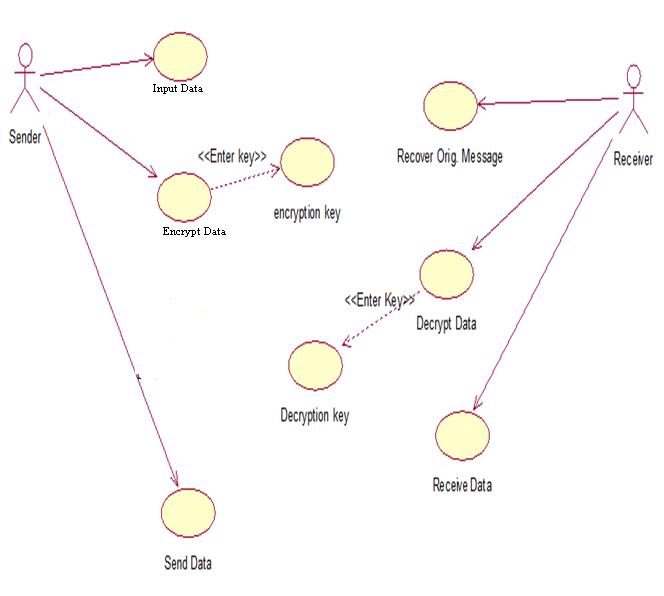
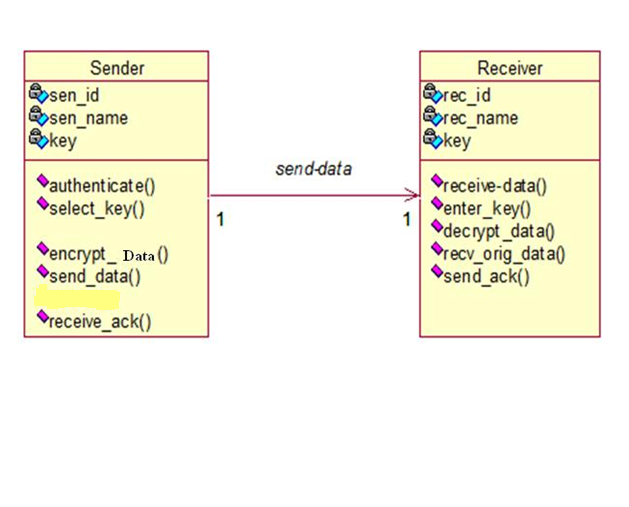


FIG 7.1.1 **USE CASE DIAGRAM FOR SENDER-RECEIVER INTERFACE**

7.2 Class Diagram:



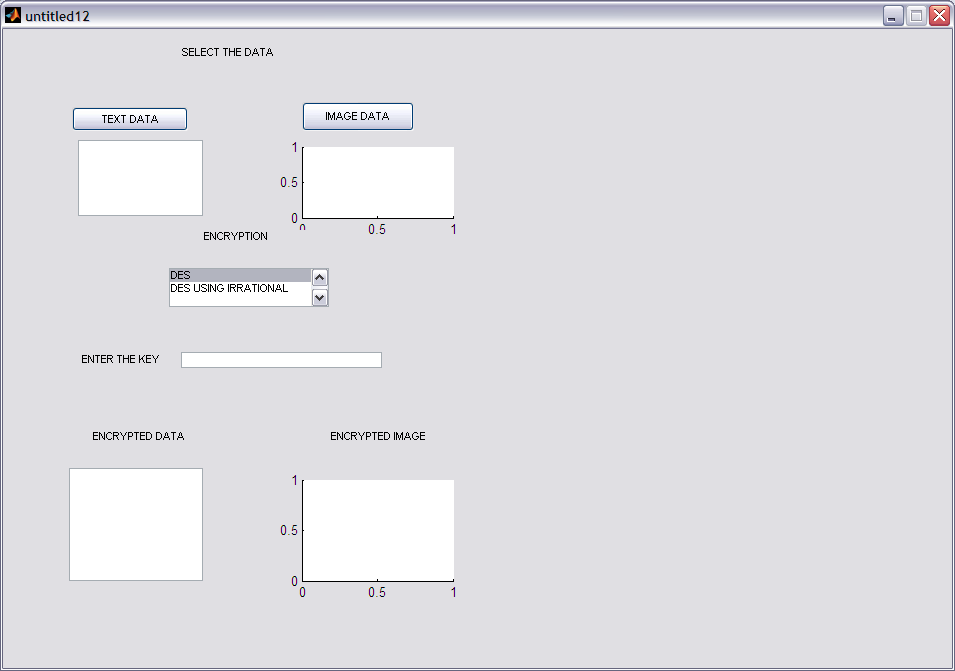
**7.2.1: Class Diagram for Sender-Receiver Communication**

7.3 Sequence Diagrams:

**Fig 7.3.1 SEQUENCE DIAGRAM FOR SENDER RECEIVER COMMUNICATION**

****

**8. SCREENSHOTS**

****

**10. CONCLUSION**

Firstly, the running time that DES took as much as that of the DES based on irrational number, i.e., the confidentiality of the key is enhanced without spending more time. Secondly, based on the same plain text and key , the cipher text of DES after several simulations is the same, but it is random about the DES controlled by irrational number, i.e., the key space is expanded through increasing the randomness of sub-keys. DES is a common encryption method. Due to the smallerkey space, the applications of DES are limited. It is our purpose that designs an encryption algorithm with better security, bigger secret-key space and higher encrypting efficiency. An improved scheme based on irrational number is proposed in this paper. Simulation results show that this scheme can expand the key space without any more running time and enhance the security of the encryption algorithm. As we can see, the algorithm has wider foreground applications based on these advantages.

10.1 Benefits:

* Secrecy
* Secure
* Flexible
* Secured
* Easy to use

10.2 Features:

* Only authorized users can use the application.
* The data to be sent can be either recorder on the spot or can be pre recorded.
* Uses Cryptographic methods of highest degree.
* No interception of the original message by any unidentified source.
* Requires the same key on the sender side and the receiver side..
* **APPENDIX**

LIST OF FIGURES:

FIGURE PAGE NOS.

USE-CASE DIAGRAM 17

CLASS DIAGRAM 18

SEQUENCE DIAGRAM 19

LEVEL-0 DATA FLOW DIAGRAM 20

LEVEL-1 DATA FLOW DIAGRAM 21

LEVEL-2 DATA FLOW DIAGRAM 22

LEVEL-3 DATA FLOW DIAGRAM 23

SCREENSHOT -1 24

SCREENSHOT-2 25

**REFERENCES**

[[1] H. Katzan, *The standard data encryption Algorithm*,People’s Posts and Telecommunications publishing house, May 1977.

[2] Y.H. Ma, “The study of cryptographic key permutation choose for enhancing DES encryption intensity”, *Computer* *Engineering*, Vol.26 No.3, pp.68-69, March 2000.

[3] C. Gu, “Dynamic DES algorithm”, *Computer Applications*

630 *and Software*, Vol.24 No.7, pp.164-166, Jul.2007.

[4] Z. Lin, “A Study And Analysis On A High Intensity Public Data Encryption Algorithm”, Proceedings of the 3d World Congress on Intelligent Control and Automation, June 28-July,

Hefei, P.R China, pp.2492-2494, 2, 2000.

[5] T. Zuo, G-P. Jiang, “Research of DES Algorithm Based on Chaotic Sequences”, *Information Security and Communications* *Privacy*, pp.33-34, Feb.2004.

[6] L.S. Wang, “Analysis and Attack Research of DES Cipher”,*Computer Engineering,* Vol.29 No.1, pp.130-132, August 2003.

[7] Y. Li, “Dynamic DES Based on S List”, *ComputerEngineering,*Vol.31 No.23, pp.127-126, December 2005.

[8] F.Y. Wang, “Dynamic Key 3DES Algorithm of Discrete System Based on Multi-dimension Chaos”, *Microelectronics &Computer,* Vol.22 No.7, pp.120-123, 2005.