*INTRODUCTION*

**1. INTRODUCTION**

* 1. **Background of the Project**

In the corporate world, each organization has as much as thousands of applications, these would have been typically written in numerous languages which in turn runs in diversified platforms. Despite of these complications, if we are still able to maintain these applications and provide the user, the best experience it is because of logging. The process of making an entry in the log file is called logging. These logs generated during the execution of an application, can be recorded in multiple destinations such as a local disk, console, into other sockets or files. Analysis of these log files is necessary later during debugging or enhancement of the respective application.

Logging is crucial for a developer, who has to modify the application, to add new functionality, or modify the existing one, which has been developed years ago. The main components of logging are: Logger, Formatter, and Appender. The Logger captures the information with metadata that has to logged. Formatter formats the object. Appenders publish the log information into a variety of ConsoleAppender, FileAppender, RollingFileAppender, and SocketAppender. Each logger can be associated with different levels of severity such as fatal, error, warning, info, debug, trace in the descending order. So a logger if configured to warning level would log all levels above it. Logger has overhead associated with it, so developer has to be judicious on what is being logged. Log manager creates the logger and manages corresponding configuration files.

In this application, extraction of log files and corresponding analysis is done. In the corporate world, each organization has its datacenters in different geographical locations far from each other, in order to withstand any natural calamities. Each datacenter in turn has numerous severs associated with it. Each application runs in some hosts of the server, produces log files as the application is being executed.

Our application helps the developer and the support team to derive the relevant files from loads of log files by specifying the severity level and the keywords, which can be converted into pie charts, bar graphs, and text files for further debugging, and analysis. This flexible modular application can handle complex queries by matching patterns and can be customized to organizational needs. This application can be invoked whenever necessary.

*SOFTWARE PROJECT PLAN*

**2. SOFTWARE PROJECT PLAN**

**2.1 Existing System**

Splunk is a log analyser tool. It provides facilities like indexing, searching, visualizing, analyzing and reporting of log data. It also provides a way to identify and solve the bugs. But the system is not customised for any organisation so it is dearth of any application specific filters.

**2.2 Proposed System**

In the proposed system, we are going to create a customised log analyser tool which can get rid of the complexity of maintaining the application specific logs. The system will be able to filter the logs in real time. The system receives the log files, generated during the user interaction with the application. The log files are analysed and the results are filtered in accordance to the user demand.

These results can be transformed into multiple formats which include excel, txt, JSON and can be sent to the support team members for further actions. The trend of the application usage can be visualised through various visual aids such as graph, pie chart, bar graph.

**2.3 Modules**

1. Magic Square Construction

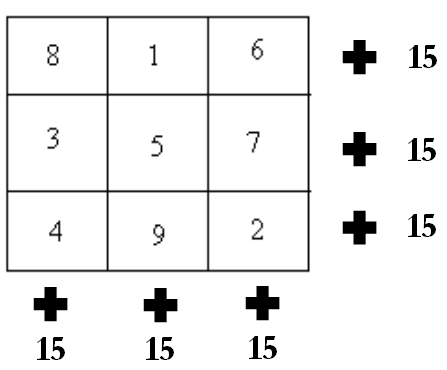
2. Strengthened Key Extraction

3. Encryption

4. Variations in Magic Square Construction

***2.3.1 Magic Square Construction***

Magic Square is a matrix of size n where the numbers from 1 to in the matrix in a manner such a way that the sum when done in the row-side or column-side or diagonal-side gives a sum of. The initial magic square is constructed using the user given key. The empty cell of magic square are filled with characters in a specific order.

******

***2.3.2 Strengthened Key Extraction***

The strengthened key is extracted from the constructed magic square by traversing through the matrix in different orders such as row-wise or column-wise. The variations include traversals such as depth-first, breadth-first, zig-zag, maze etc.

***2.3.3 Encryption***

Once the strengthened key is extracted, it is fed as an input to any cryptographic algorithm which is to be enhanced in terms of security. The cipher text obtained from the encryption is highly secured in a way such that it makes the intruders almost impossible to crack the data transmitted.

***2.3.4 Variations in Magic Square Construction***

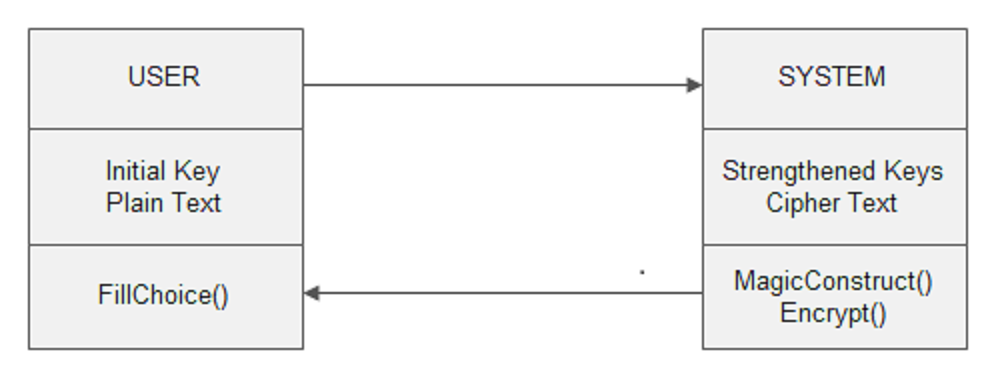
The basic method of constructing magic square includes filling the empty cells using alphabets from A to Z or in reverse order. The other variations include filling the empty cells with numeric characters or complex manipulations such as XOR of two magic squares.

*SOFTWARE REQUIREMENT SPECIFICATION*

**3. SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 Functional Requirements**

***3.1.1 Class Diagram***

******

*Fig. no: 3.1.2 Class Diagram for user and system.*

***3.1.2 Front End Application***

In this project, Command Prompt is used to get the initial key from the user, choice of filling empty cells in magic square and the input file to be encrypted.

**3.2 NON FUNCTIONAL REQUIREMENTS**

***3.2.1 Performance Requirements***

The objective is to provide an efficient way of encryption to improve security in digital transmission of data. The user given key is strengthened and used as input for any cryptographic algorithm. The output of encryption, which is the cipher text, is stored in form of files. This file is again used for decryption of the cipher text to obtain the original message or the plain text.

***3.2.2 Interface Requirements***

The command prompt is used as an interface between the user and system where the user enters the initial key and the input file or text to be encrypted is given.

***3.2.3 Resource Requirements***

* *Software Requirements:*

Operating System - Windows

Front End - Java Applet

Language - Java

* *Hardware Requirements:*

Processor - Pentium IV or more

RAM - 512 Mb or more

Hard Disk - 50GB or Higher

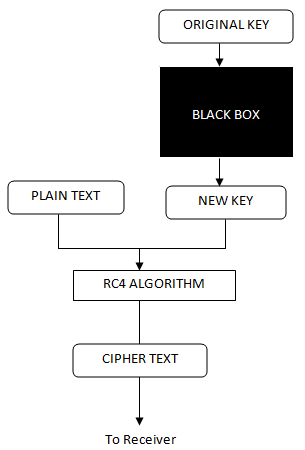
Monitor - Display Panel(640 x 480)

*SYSTEM ANALYSIS*

**4. SYSTEM ANALYSIS**

**4.1 Dataflow Diagram**

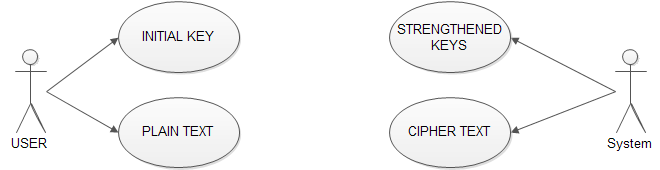
A data-flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. DFDs can also be used for the visualization of data processing.



*Fig. no: 4.1.1 Dataflow diagram.*

**4.2 Use Case Diagram**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.



*Fig. no: 4.2.1 Use Case diagram*

*SYSTEM DESIGN*

**5. SYSTEM DESIGN**

**5.1 Front End Design**

***5.1.1. Applet***

***5.1.2. Java (Programming Language)***

***5.1.3. Versions***

***5.1.4. Java platform***

***5.1.5. Implementations***

***5.1.6. Performance***

***5.1.7. Automatic memory management***

***5.1.8. JAVA (Software Platform)***

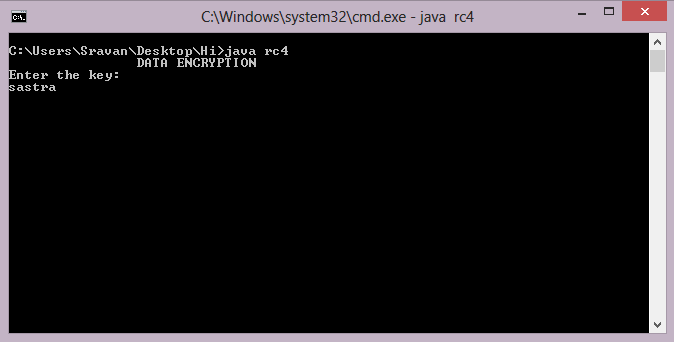
***5.1.9. Java Virtual Machine***

*CODING*

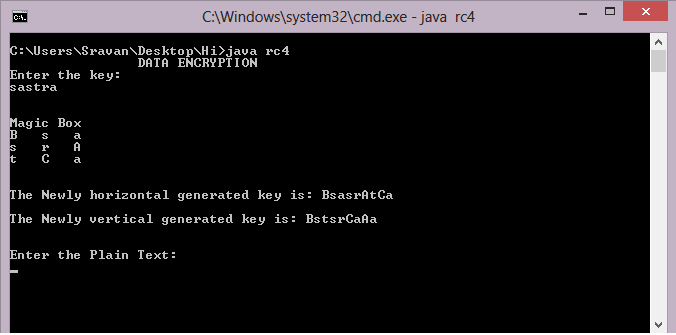
**6. CODING**

**6.1 Sample Coding**

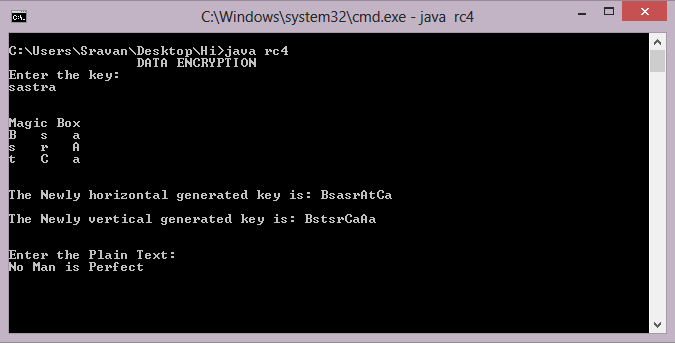
**6.2 Screenshots**

**

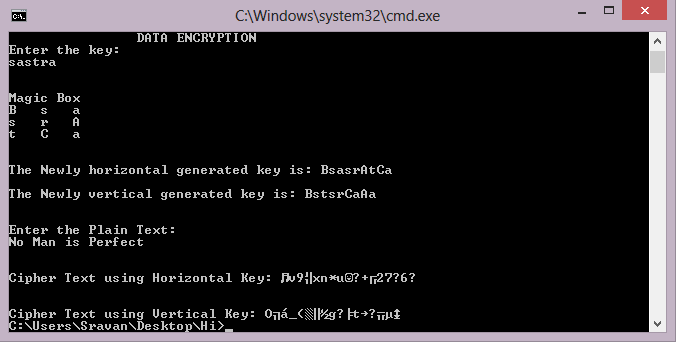
*Fig. no: 6.2.1 Getting the initial key from the user*

**

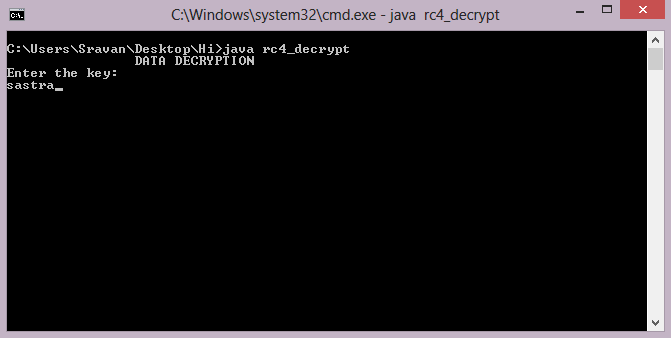
*Fig. no: 6.2.2 Magic Square Construction and Strengthened Generation*

**

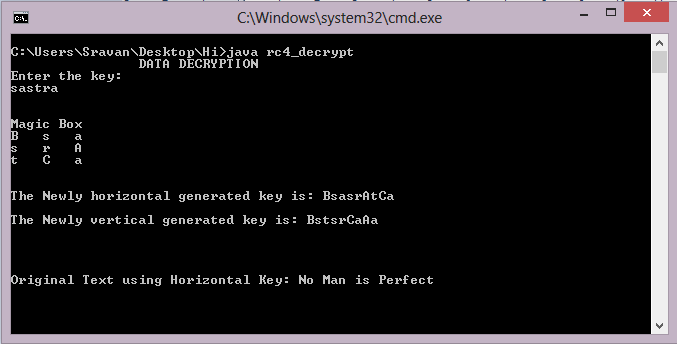
*Fig. no: 6.2.3 Getting the Plain Text from the User*

**

*Fig. no: 6.2.4 Generation of Cipher Text*

**

*Fig. no: 6.2.5 Getting the key from the user for Decryption*

**

*Fig. no: 6.2.6 Decryption of Original Message*

*IMPLEMENTATION*

**7. IMPLEMENTATION**

**7.1 Problems Faced**

The main objective is to provide purely random and dynamic keys for encryption. The process of traversing the magic square to extract different patterns of keys was difficult. The implementation of encryption algorithms using various keys simultaneously was tedious. Generating a highly secured and uncrackable cipher text was challenging.

**7.2 Lessons Learnt**

Through this project, we learnt in depth knowledge about cryptography and importance of secured communication given the advancement in hacking techniques. We learnt about the concepts of cryptanalysis and tried cracking out cipher text using brute force attack tools. We also learnt how important are the communications that happens between two parties nowadays.

*FUTURE SCOPE*

**8. FUTURE SCOPE**

In our proposed system, the order of Magic Square we use is ODD. This approach can also be implemented using Magic Squares of EVEN order. Here we use only certain patterns of key extraction. These patterns can be extended to further complex ways using several techniques computer graphics, where we can use pixel positions to extract the key from the magic square.

*REFERENCES*

**9. REFERENCES**

[1]G. Manikandan, R. Manikandan, G. Sundarganesh,2011, *A New approach for generating Strong Key in RivestCipher4 Algorithm*, Journal of Theoretical and Applied Information Technology,113-119.

[2]G.Manikandan, N.Sairam, M.Kamarasan, 2012,*A New Approach For Improving Data Security Using Iterative Blowfish Algorithm,*Research Journal of Applied Sciences, Engineering and Technology, 4(6) ,,603-607.

[3]G.Manikandan, N.Sairam, M.Kamarasan, A Hybrid Approach For Security Enhancement by Compressed Crypto-Stegno Scheme in Research Journal of Applied Sciences, Engineering and Technology,Vol 4(6) ,2012,

608-614.

[4] B.Karthikeyan, V.Vaithiyanathan, B. Thamotharan, M.Gomathymeenakshi and S.Sruthi, LSB Replacement Stegnography in an image Using Psudorandomised Key Generation. Research Journal of Applied Sciences, Engineering and Technology, 4(5), 2012, 491-494.

[5]S.Karthikeyan, N.Sairam, G.Manikandan, J.Sivaguru, A Parallel Approach For Improving Data Security in Journal of Theoretical and Applied Information Technology ,Vol 39(2) , May 2012,119-125.

[6]G.Manikandan, P.Rajendiran, K.Chakarapani, G.Krishnan,G. SundarGanesh, A Modified Crypto Scheme for Enhancing Data Security in Journal of Theoretical and applied information Technology ,Vol 35(2) , January 2012,149-154.

[7]S.Karthikeyan, N.Sairam, G.Manikandan, A New Approach For Enhancing Data Security Using Parallel Processing in Advances in Natural and Applied Sciences ,Vol 6(6) , July 2012,696-703.

[8]G.Manikandan,N.Sairam,M.Kamarasan, A New Approach for Secure Data Transfer based on Wavelet Transform in International Journal of Network Security, Vol 15(1) , 1-March 2013,88-94.