Journal of Advances in Science and Technology
Vol. 19, Issue No. 1, March-2022, ISSN 2230-9659

File Security System

Tathe Deshmukh B. A.[1]* Kale S. D.[2] Ms. Sawant V. N.[3]

[1,2] TY Diploma Students, Department of Computer Engineering, Sahakar Maharshi Shankarrao Mohite Patil Institute of Technology and Research, Akluj, Solapur, Maharashtra, India

[3] Lecturer, Department of Computer Engineering, Sahakar Maharshi Shankarrao Mohite Patil Institute of Technology and Research, Akluj, Solapur, Maharashtra, India

Abstract – We primarily demonstrate how to store files securely using encryption methods in this File Security System application. The user will log in to the app by providing a valid email address to which the file security key will be provided. After a successful login, the user will upload a file, which will be encrypted and stored in the specified directory, with the security key being delivered to the specified email address. By entering the security key that was sent to the user's email address, the user will be able to download the encrypted file.

Key Word – Secure System, Secure Files Are Just a Few of the Terms That Come to Mind When Thinking About.

----------X----------

INTRODUCTION

In the government, the commercial sector, and other types of organizations, there is a vast deal of secret information about their personnel, customers, products, research, and financial situation. Almost all information is now collected, processed, and stored on electronic computers, and then communicated to other computers via networks. If sensitive information about a company's clients, finances, or new product line falls into the hands of a competitor, the company could lose sales, face legal action, or possibly go bankrupt. Protecting it is a business necessity, as well as an ethical and legal one in many circumstances. Information security has a substantial impact on individual privacy, which is seen differently in different cultures. All information is stored in computer systems**.**

LITERATURE REVIEW

The Data Encryption Standard (DES) is a symmetric-key encryption technique for electronic data. Despite the fact that it is today regarded insecure, it had a significant impact on the development of current cryptography. [2] The algorithm, which was created at IBM in the early 1970s and was based on an earlier concept by Horst Feistel, was submitted to the National Bureau of Standards (NBS) in response to a request to offer a candidate for the security of sensitive, unclassified electronic data from the govt. The NBS eventually chose a slightly modified version (stronger against differential cryptanalysis but weaker against brute force attacks) in 1976 after consulting with the National Security Agency (NSA), which was published as an official Federal Information Processing Standard (FIPS) for the United States in the year 1977

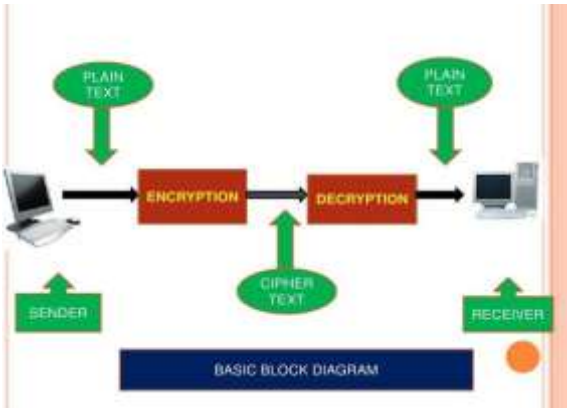
METHODOLOGY

Traditionally, software complexity has been used to determine software quality and cost. The more complicated the software (in some ways), the more prone it is to failure, resulting in higher costs. One of the simplest things to gauge is the size of a code. The size of the code can be used to forecast software qualities like effort and upkeep. A line of code (LOC) is a unit of measurement for software complexity.

Many meaningful comparisons are limited to the number of lines of code in a project's order of magnitude. The number of lines of code in a software project might range from one to a billion or more. Comparing a 10,000-line project to a 100,000-line project using lines of code is considerably more useful than comparing a 20,000-line project to a 100,000-line project, software complexity has been used to determine software quality and cost. The more complicated the software (in some ways), the more prone it is to failure, resulting in higher costs. One of the simplest things to gauge is the size of a code. The size of the code can be used to forecast software qualities like effort and upkeep. A line code (LOC) is a unit of measurement for software complexity.

Many meaningful comparisons are limited to the number of lines of code in a project's order of

magnitude. The number of lines of code in a software project might range from one to a billion or more. Comparing a 10,000-line project to a 100,000-line project using lines of code is considerably more useful than comparing a 20,000-line project to a 100,000-line project.



CONCLUSIONS

A new simple tool has been created which is targeted from use inside of small institution such as a small university of lecturers daily use of sending exam file an sensitive material such that the material can be encrypted and the file is send one Email while the encrypted key is send in another Email or via any secure communication channel.

REFERENCES

[1] T. P. Singh and R. K. Tuteja. Brijender Kahanwal, T. P. Singh, and R. K. Tuteja — A Windows Based Java File Security system (JFSS)".

[2] Vol. 976-8491 of the Vol. 976-8491 of the Vol. 976-8491 of the International Journal of Computer Science and Technology (ISSN: 0976-8491).Vol. 2, Issue 3, pages 25-29, September 2011.

[3] M. Blaze (1993). A Cryptographic File System for UNIX, in ACM Conference on Computer and Communications Security, pages 9-16.

[4] G. Cattaneoand G. Persiano. — The second extended file system was designed and implemented. Systemfor UNIX, In Proceedings of the Annual USENIX Technical Conference, REENIX Track, pages 245-252, June 2001.

[5] EncFS [Online] Available: http://www.arg0.net/encfs & http://encfs.sourceforge.net/

[6] Ext3 [Online] Available: http://www.zipworld.com.au/~akpm/linux/ext3/

[7] H. Reiser. Reise rFS. [Online] Available: http://www.namesys.com/. & http://www.reiserfs.org

[8] R. Card, T. Ts'o, and S. Tweedie (1994). Design and implementation of the second extended file system. In Proceedings to the First Dutch International Symposium on Linux, Seattle, WA.

[9] A. Sweeny, D. Doucette, W. Hu, C. Anderson, M. Nishimoto, and G. Peck (1996). Scalability in the XFS File System, In USENIX.

[10] T. P. Singh and R. K. Tuteja. Brijender Kahanwal, T. P. Singh, and R. K. Tuteja (2011). Pelagia Research Library—Advances in Applied Science Research (ISSN: 0976- 9610), Volume 2, Issue 6, PP--254-260.

[11] T. P. Singh and R. K. Tuteja. Brijender Kahanwal, T. P. Singh, and R. K. Tuteja —Towards a Framework for the Performance Evaluation of File Systems and the.

Corresponding Author

Tathe Deshmukh B. A.*

TY Diploma Students, Department of Computer Engineering, Sahakar Maharshi Shankarrao Mohite Patil Institute of Technology and Research, Akluj, Solapur, Maharashtra, India