

DATA SECURITY MODEL FOR ROLE –BASED ACCESS CONTROL UTILIZING (RSA) ALGORITHM

Ugochukwu Chikadibia Emmanuel¹, Prince Oghenekaro Asagba², Friday Eleonu Onuodu³

^{1,2,3} Department of Computer Science, University of Port Harcourt Choba, Nigeria

*Corresponding Author: chikamobil100@gmail.com

D.O.I.: 10.5281/zenodo.20398013

ARTICLE INFORMATION

Received: 17th March, 2026

Accepted: 15nd April, 2026

Published: 25th May, 2026

KEYWORDS: Algorithm, Role-Based, Control, Security, Computing

Publisher: Empirical Studies and Communication (A Research Center)

Website: www.cescd.com.ng

ABSTRACT

In today's computing era, every business thrives on sensitive, valid and crucial data which entails processes that need to be secured. The existing models falls short in addressing the issue of unauthorized user gaining access to confidential files as well as central server, without the admin approval. This study addressed the security challenges in access control for data security in the computing environment. The model training was done with the agent parameters by utilizing Rivest, Shamir, and Adelman (RSA) algorithm which led to obtaining a more precise and optimal value. The model entails the use of the principles of Role –Based Access Control and refers to the idea of assigning permission to user based on their role within an organization. The new model provides availability of database and create large space memory location that can handle bulk unstructured information in the computing environment. The proposed system is built using Object Oriented Design Approach (OODA), JavaScript and Python Programming Language is employed as the backend while HTML and CSS was used as the frontend and MYSQL for relational database while flask were used to develop the web-based user interface. Different sizes of files were uploaded to the distributed sever and time for execution (encryption and decryption) of these files were determined to indicate the amount of time taken to encrypt each file depending on its sizes. The times taken in millisecond for the proposed system are: 246 ms, 158ms, 362ms, 191ms and 783ms while the file size in megabytes are: 46.19mb, 116.56mb, 188.83mb, 302.25mb, and 6858.25mb respectively. The results in study show that the proposed model was tested and performed efficiently, by taking lesser time to execute files than other existing models and can also guarantee data security for role-based access control in a Computing environment.

1.0 Introduction

A secure and trusted data allotment in a distributed environment is a very critical research design approach. At present, there are new threats damaging information systems and data resources thus, security is a goal in every organization [2]. In today's computer driven world, every business has some sensitive, crucial data and processes that need to be secured. Combining the various definitions of many researchers, it can be stated that a distributed

system is an application that communicates with multiple dispersed hardware and software in order to coordinate the actions of multiple processes running on different autonomous computers over a communication network, so that all components hardware and software cooperate together to perform a set of related tasks targeted towards a common objective [4]. Distributed system has features over a centralized system such as economics, speed, inherent distribution, and incremental growth. . Vulnerabilities of computer systems range from the possibility of a trusted employee's selling (or being forced to reveal) secrets to a competitor, disk failures that produce system crashes, unauthorized operating system penetration, inferences about confidential data through carefully chosen queries posed to a database (DB), loss of data due to natural causes, acquisition of data through wiretapping, denying access to the database by flooding the system, or harming the databases with malicious software[3].

2.0 Related Review

Security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and valuable asset, such as a person, dwelling, community, item, nation, or organization [1]. Security provides "a form of protection where a separation is created between the assets and the threat." These separations are generically called "controls," and sometimes include changes to the asset or the threat. Security has two dialogues. Negative dialogue is about danger, risk, threat and etc. Positive dialogue is about opportunities, Interests, profits and etc. Negative dialogue needs military equipment, armies, police. Positive dialogue needs social capital, Education, social interaction. The objective of a trustworthy computer system in a distributed system is to control access by subjects (users) to objects (data). This control is governed by a set of general goals and objectives called a security policy [5].

2.1 Security Policy

A security policy is a set of rules and practices dictating how sensitive information is managed, protected, and distributed. A security policy expresses exactly what the security level should be by setting the goals of what the security mechanisms are to accomplish [7]. This is an important element that has a major role in defining the design of the system. The security policy is a foundation for the specifications of a system and provides the baseline for evaluating a system. Security policies were examined in depth, but those policies were directed toward the company itself [11, 12]. The security policies being addressed here are for operating systems and applications. The different policies are similar but have different targets: an organization as opposed to an individual computer system [6]. A system provides trust by fulfilling and enforcing the security policy and typically deals with the relationships between subjects and objects. The policy must indicate what subjects can access individual objects, and what actions are acceptable and unacceptable. The definition of what trust means is derived from a framework and the security policy works as this framework for computing systems. For a system to provide an acceptable level of trust, it must be based on an architecture that provides the capabilities to protect itself from untrusted processes, intentional or accidental compromises, and attacks at different layers of the system [13].

2.1.1 Security Models

An important concept in the design and analysis of secure systems is the security model, because it incorporates the security policy that should be enforced in the system [8]. A model is a symbolic representation of a policy. It maps the desires of the policy makers into a set of rules that are to be followed by a computer system. We have continually mentioned the security policy and its importance, but it is an abstract term that represents the objectives and goals a system must meet and accomplish to be deemed secure and acceptable [9]. How do we get from an abstract security policy to an administrator being able to uncheck a box on the

graphical user interface (GUI) to disallow David from accessing configuration files on his system? There are many complex steps in between that take place during the system's design and development. A security model maps the abstract goals of the policy to information system terms by specifying explicit data structures and techniques necessary to enforce the security policy [10].

3.0 Proposed System

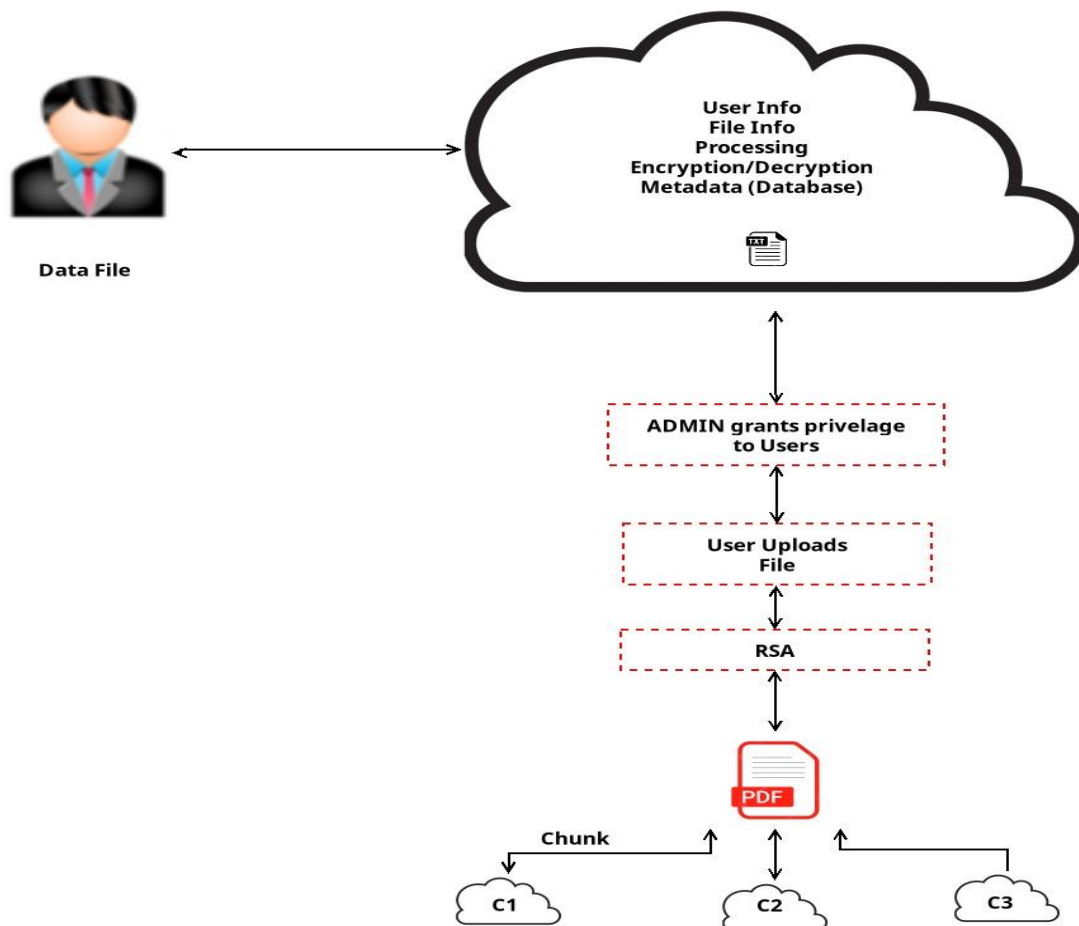


Figure 3.1: Proposed System Architecture

3.1 Modules Description

The Proposed System components include:

Data File: Users upload data to servers via an internet connection, where it is saved on a virtual machine on a physical server. To maintain availability and provide redundancy, cloud providers will often spread data to multiple virtual machines in data centers located across the world. Users can access data in Cloud Storage through an internet connection and software such as web portal, browser, or mobile app via an application programming interface (API).

File Information: The file information is the collection of data stored in the distributed system. Basically, these files are used for storing information about users, software applications, or other data that are needed for running the programs.

Processing: Processing system enables to read, write, modify, and store data. This involves different transactions performed by the user. Some of the transaction includes, Definition of a

new transaction, Deletion of existing transaction, and Updating of the transaction details, Receiving the list of transactions authority and Receiving authority for transaction.

Encryption/Decryption: Encryption is implemented to convert readable message to an unreadable form to in other prevent unauthorized parties from reading it. Decryption converts an encrypted message back to its original (readable) format.

Database Base: This is where all individual files are stored. Encryption of transaction is applied to each user file and transactions performed.

Role-Based Access Control: The roles of the users are grouped based on their common responsibilities. Each user is assigned with one or more roles and one or more permissions to each role. The user role and role permissions relationship make it simple to perform user assignments since user no longer need to be managed individually, but instead have privileges that conform the permissions assigned to their role.

3.1.1 Algorithm for the Proposed System

In developing the efficient model for data security in a distributed environment, Rivest, Shamir, and Adelman (RSA) algorithm was applied. The algorithm is based on asymmetric encryption and decryption principle. In this algorithm Public key is distributed to all through which one can encrypt the message and private key which is used for decryption is giving and kept secret and is not shared to everyone. Set of Functions in the Algorithm are:

NumberOfBlock(F) : It returns the number of block in the file F.

ENC_AES (B,K) : It encrypts the block B with key K.

send_to_cloud(F') : It permits to send the encrypted file F in Cloud storage

ENC_RSA(k) : It encrypts k using RSA Algorithm.

Save_in_server(K') : It permits to save K' in the server

NumberOfBlock(F) : It returns the number of block in the file F.

DEC_RSA(k') : It decrypts k' using RSA Algorithm.

DEC_AES(B',K) : It decrypts the block B with key K.

3.1.2 Algorithm 1: Algorithm for File Upload

Step 1: Encrypt_file (F)

Step: 2 Obtain public key of recipient $Pu=\{n, e\}$ to calculate the cipher: $C=M^{e \bmod(n)}$, where $0 \leq M < n$.

Step 3: To transform Clair text in file F into Cipher text in file F'. Set $a^{\phi(n)} \bmod(n) = 1$ where $\gcd(a,n)=1$ and calculate $n=p.q$ such that $\phi(n)=(p-1)(q-1)$ then chose e and d to be inverses mod $\phi(n)$.

Step 4: For $B \leftarrow 1$ to numberOfBlock(F) do

Step 5: Else

Step 6: $B' = \text{ENC_AES}(B, K)$

Step 7: else

Step 8: send_to_cloud(F')

Step 9: For $k \leftarrow 1$ to SizeOf(K) do

Step 10: else

Step 11: $k' = \text{ENC_RSA}(k)$

Step 12: decryption of the recipient uses their private key $Pr=\{n, d\}$ and computes: $M=C^{d \bmod(n)}$.

Step 13: end for statement
Step 14: Save_in_server(K')
Step 15: Exist

3.1.3 Algorithm 2: Algorithm for File_Download

Step 1: Decrypt_file (F')
Step 2: obtained private key $Pr = \{n, d\}$ and computes: $M = C^d \text{mod}(n)$.
Step 3: transform Cipher text in file F' into Clair text in file F
Step 4: For $k' \leftarrow 1$ to SizeOf(K') do
Step 5: Else
Step 6: $k = \text{DEC_RSA}(k')$
Step 7: else
Step 8: return(K)
Step 9: apply Phase 2: Decrypt Cipher text
Step 10: For $B' \leftarrow 1$ to numberofBlock(F') do
Step 11: else
Step 12: $B = \text{DEC_AES}(B', K)$
Step 13: end of statement
Step 14: return(F)
Step 15: Exist

4.0 Results

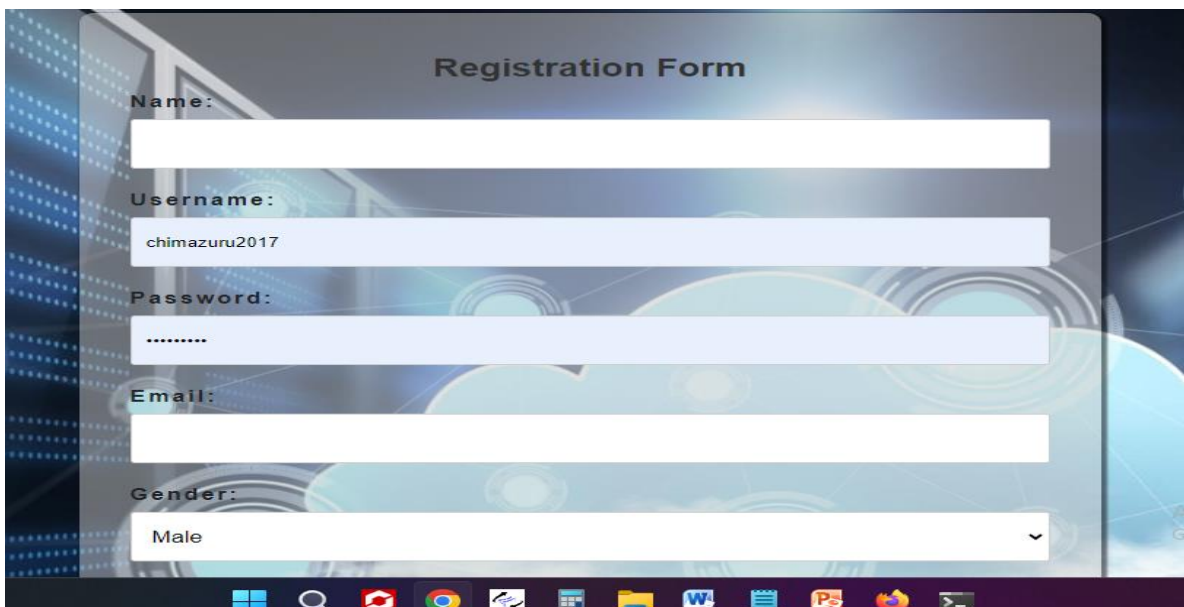


Fig 4.1: User Registration Page



Fig 4.2: Admin Dashboard

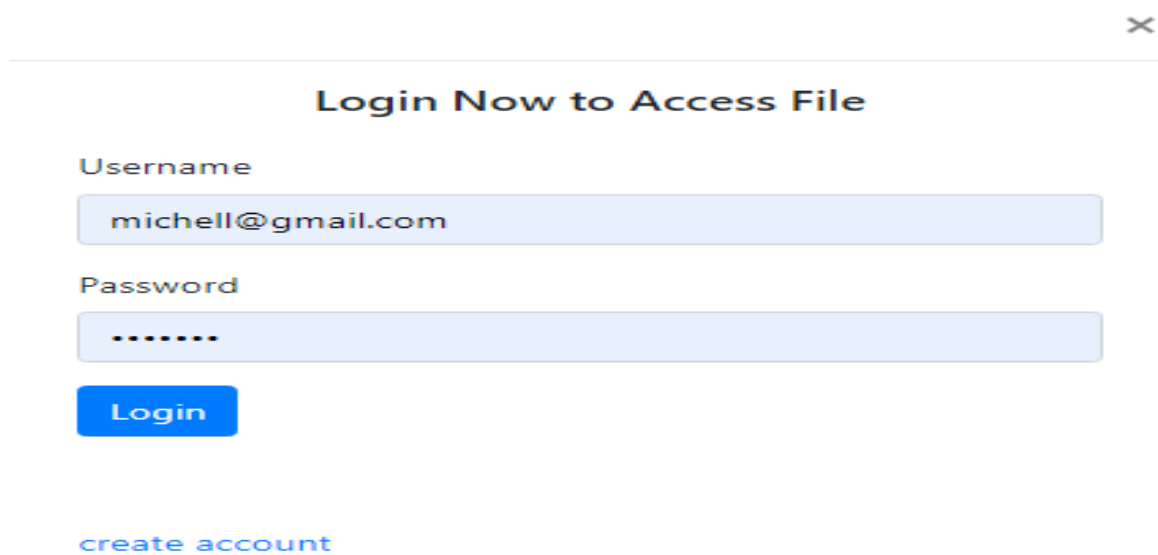


Fig 4.3: Security Code Page

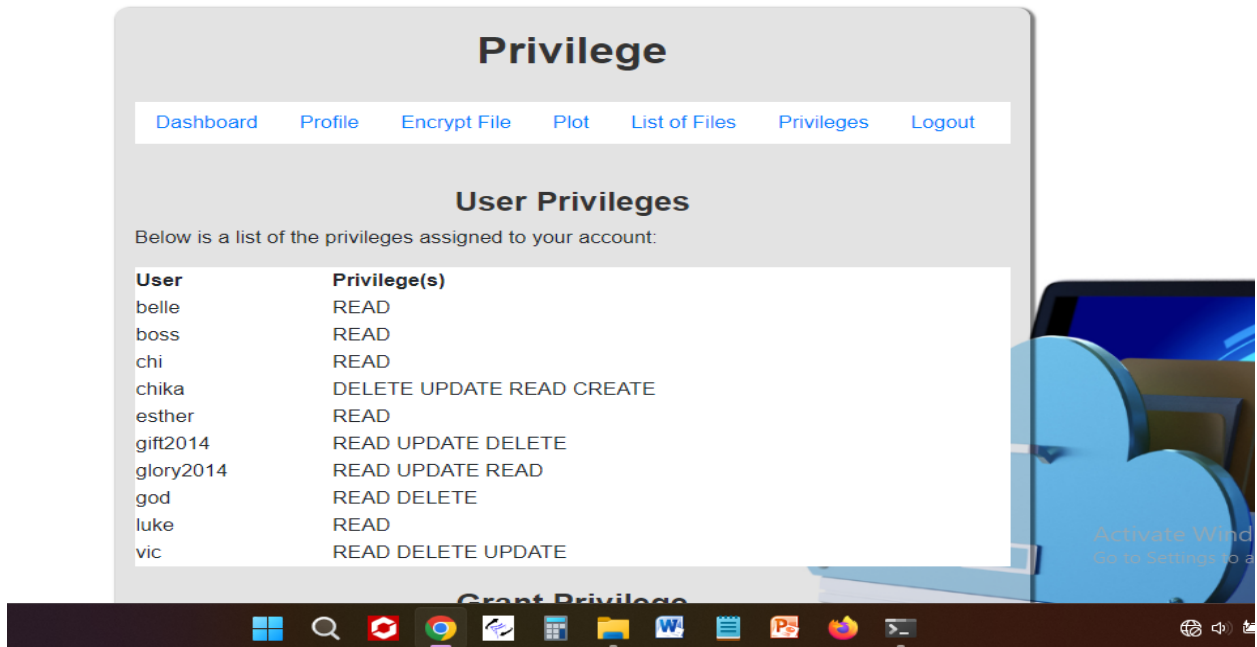


Figure 4.4: User Privileges



Figure 4.5: Encrypted PDF file

Table 4.1: Result of the Proposed System Evaluation Performance

S/N	File Size(MB)	Total Time (ms)
1	46.19	246
2	116.56	158
3	188.83	362
4	302.25	191
5	6858.25	783

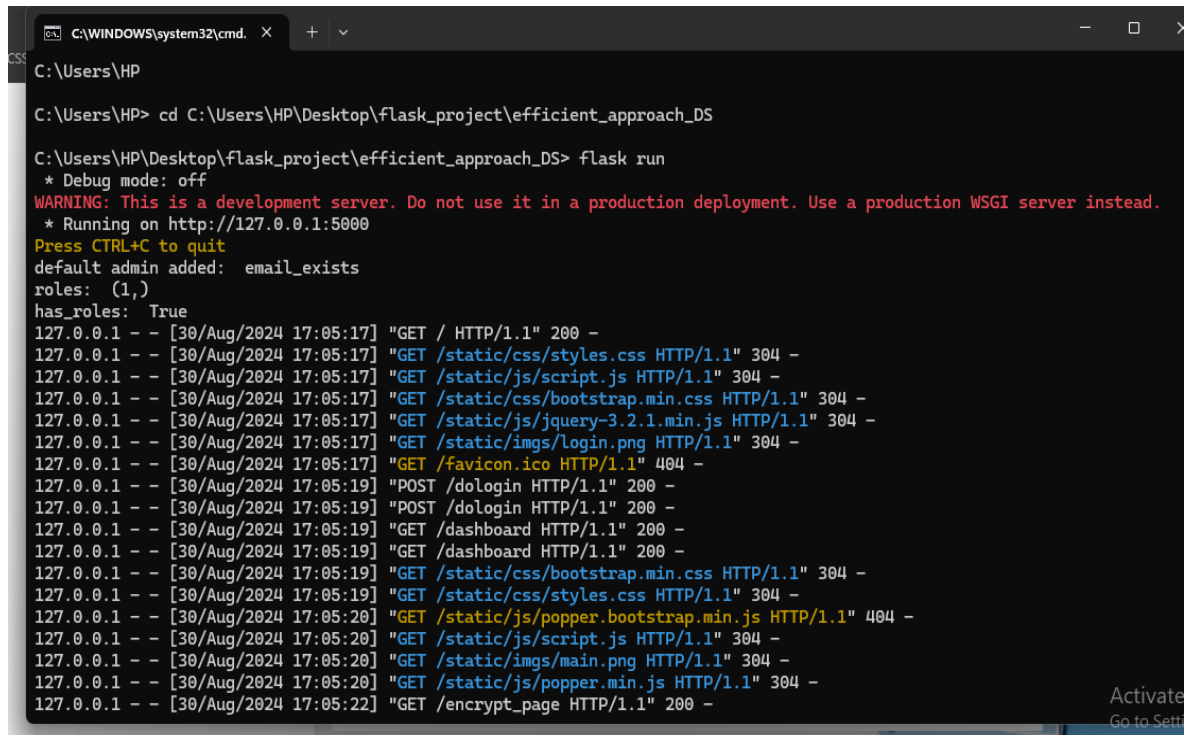


Figure 4.6: Python Flask Control Panel

Table 4.2: Test Case 1.1 (Security Key)

Test Case Number	1.1
Case Name	Security Key
Precondition	The security key is chosen by the user and apply in the application to be authenticate by the system
Case Input	create an account and enter the correct security code for login before clicking Register submit
Case Expected Output	Successful result of the key are displayed
Case Steps/Description	Input: create an account and enter the correct security code Output: Displayed successful authentication security code

Table 4.3: Test Case 1.2 (Access Files)

Test Case Number	1.2
Case Name	Access Files
Precondition	File to be access have been selected
Case Input	Select all the files to be access before clicking open file
Case Expected Output	Successful result of displayed files will be open
Case Steps/Description	Input: select file for accessing Output: Displayed updated and modify file

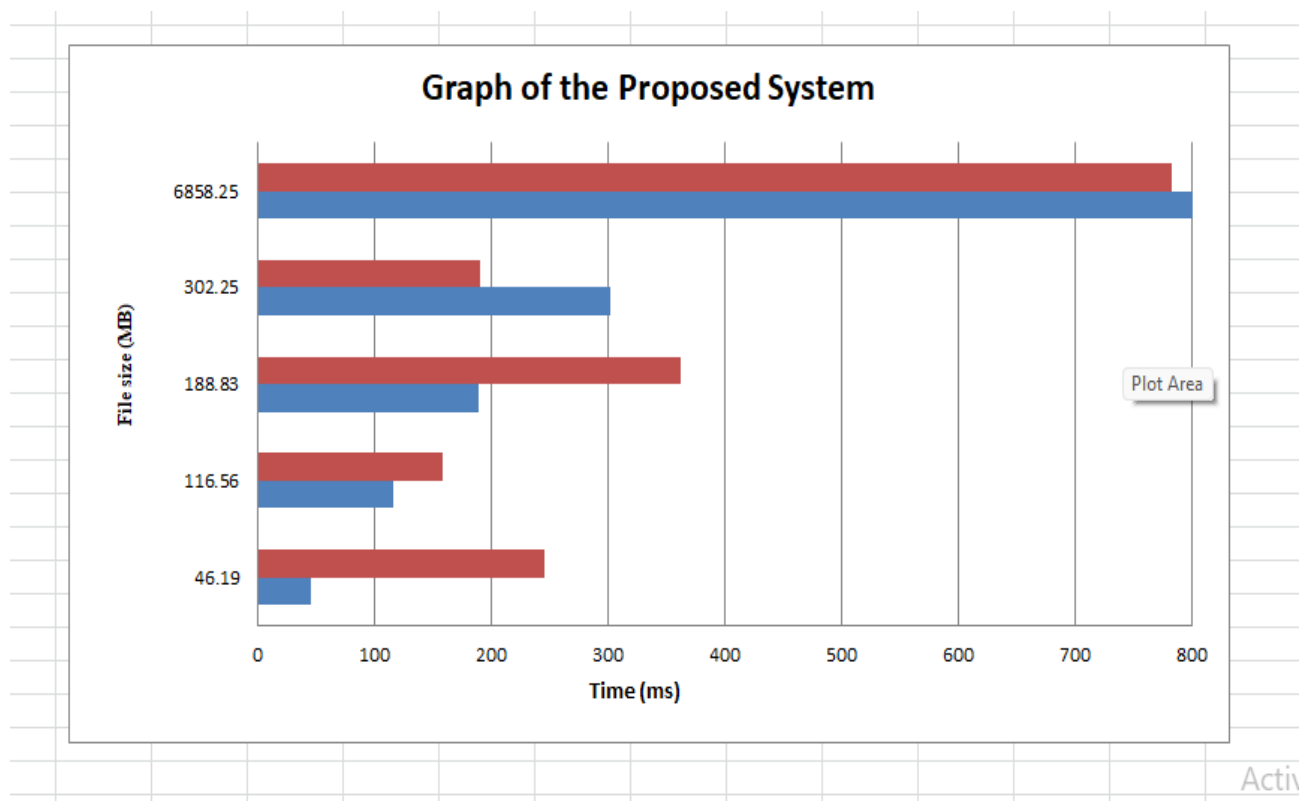
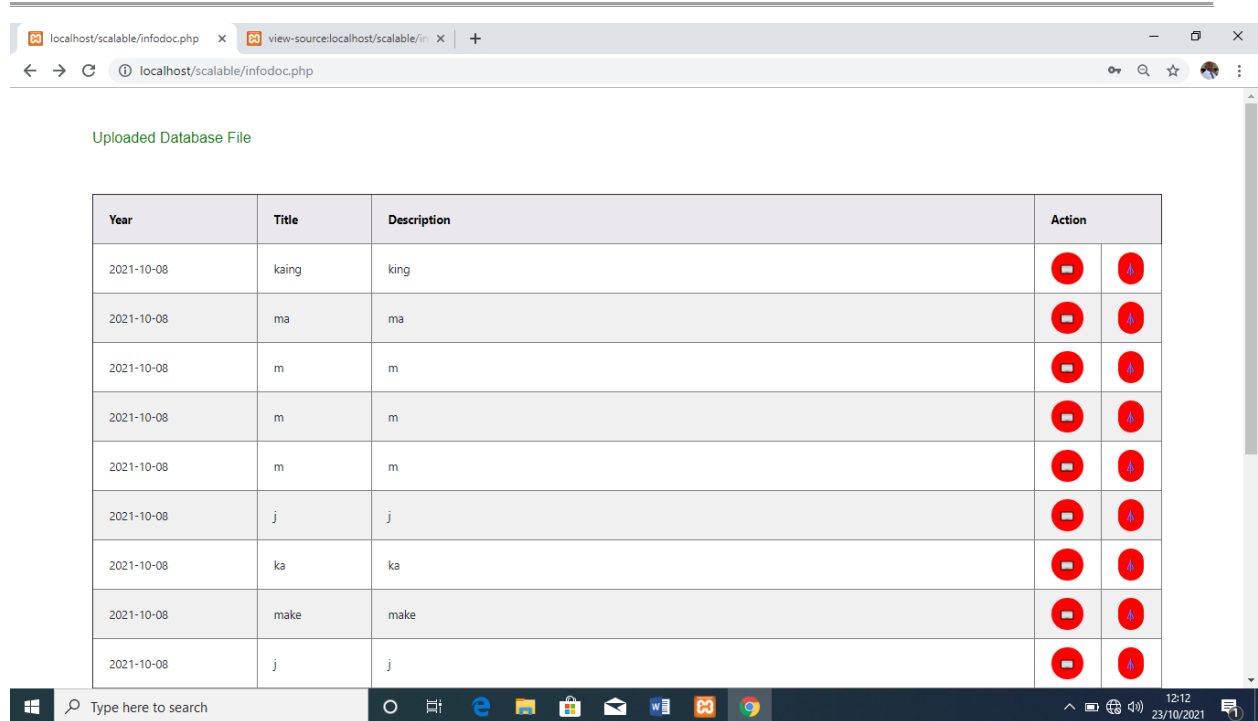


Figure 4.7: Proposed System Graph Evaluation



4.8: Uploaded Database File

5.0 Conclusion

The various challenges associated with core security principles in terms of confidentiality, data integrity and availability of memory space especially for role-based access control for file processing has been resolved in study. It was ensured that system and important data transmission between parties or different uses in a computing environment is kept completely undisclosed to the unauthorized entities or owner. For the newer architectures, such as wireless and peer-to-peer, this challenge becomes more interesting due to factors such as lack of structure and lack of trust in the network. This has now reduced the risk operational disruption, financial losses, legal issues, compliance penalties and reputational damages in new model. Maintaining the accuracy and consistency of data over its entire life-cycle is very paramount and important. Going further, hackers will no longer be able to temper and modify data in an undetected manner without permission of its legitimate user, which initially leads to data breach as the idea of role based access control, deals with assigning permission to user based on their role within an organization. The new data security model has been tested and performed efficiently in proffering solution to the security challenges in computing environments using role-based access control for file processing.

REFERENCES

[1] Azgomi, H., & Sohrabi, M. K. (2022). A game theory-based framework for materialized view selection in data warehouses. *Engineering Applications of Artificial Intelligence*, 71, 125-137.

[2] Baiao, F., Mattoso, M. & Zaverucha, G. 2024. A distribution design methodology for object DBMS. *Journal of Distributed and Parallel Databases*. 16 (1), 45-90.

-
- [3] Coulouris R. (2019). Enhanced information security in distributed mobile system based on delegate object model, *Procedia Engineering*, 34(30), 774-781.
- [4] Dhinakaran, V. (2020). StealthDB; a Scalable Encrypted Database with full SQL query Support, *Proceedings on Privacy Enhancing Technologies*; 2019(3), 370 – 388.
- [5] Elamasri, W. (2023). *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 23(43): 123-231.
- [6] Guynes (2021). Database security in a client/server environment. an access security system operating only in a local area network environment (LAN) with many stations and users. *Journal of Distributed and Parallel Databases*. 21(32): 124-541.
- [7] Ishtiaq Ahmed, M. Rizwan Beg, Kapil Kumar Gupta, Mohd.IshaMansoori,(2021). A novel approach of query optimization for genetic population, *IJCSI International Journal of Computer Science*. 9, 10-21.
- [8] Markham (2025). Next-generation access control for distributed control systems, *International Journal of Computer Applications (IJCA)*, 10(9), 47 – 52
- [9] Nihalani, A. (2024). “Integration of Artificial Intelligence and Database Management System: An Inventive Approach for Intelligent Databases”. *First International Conference on Computational Intelligence, Communication Systems*.
- [10] Pourqasem, S. (2022). Distributed database management systems and the data grid. *Journal of Computer Science and Informatics (JCSI)*, 4(7), 21 - 28
- [11] Srinivasa, D. (2019). Introduction to reliable and secure distributed programming. *European Journal of operations research*, 12(25): 14- 47.
- [12] Yong, Y. (2023). The cougar approach to in-network query processing in sensor networks. *International Journal of Computer Applications (IJCA)*, 10(9), 44 – 49
- [13] Zhao (2022). Collaborative signal and information processing.queries, *International Journal of Scientific and Technology Research (IJSTR)*, 4(10), 22-26.