



CipherCore Security Suite: A Secure Web-Based File Encryption and Management System

DIVYA SHREE B M

Department of CSE
VTU University
Belagavi, India

divyabm53@gmail.com

GOPALAKRISHNA H R

Department of CSE
VTU University
Belagavi, India

gkgowda235@gmail.com

LIKHITH KUMAR T M

Department of CSE
VTU University
Belagavi, India

likhithkumartm2005@gmail.com

RAKESH B T

Department of CSE
VTU University
Belagavi, India

rteli7185@gmail.com

Guide : Ramya MG

Asst.professor CSE department VTU University Belagavi,

India Ramyang199@gmail.com

Abstract—

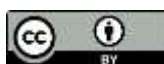
How to Cite this Article:

M, D. S. B., R, G. H., M, L. K. T. & T, R. B. (2026).
CipherCore Security Suite: A Secure Web-Based
File Encryption and Management System.
International Journal of Creative and Open
Research in Engineering and Management,
<i>02</i>(10), 1-9.
<https://doi.org/10.55041/ijcope.v2i10.083>

License:

This article is published under the terms of the
Creative Commons Attribution 4.0
International License (CC BY 4.0), which
permits unrestricted use, distribution, and
reproduction in any medium, provided the
original author(s) and the source are credited.

© The Author(s). Published by International
Journal of Creative and Open Research in
Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i10.083>

The growing use of digital technology has made file sharing, online storage, and electronic data management an essential part of everyday life, but protecting sensitive information from unauthorized access, data theft, and misuse remains a major challenge. Traditional file management methods may not effectively protect confidential documents, prevent unauthorized access, or ensure the secure handling of sensitive data. To address these challenges, this paper presents CipherCore Security Suite: A Secure Web-Based File Encryption and Management System that integrates file encryption, user authentication, password hashing, email-based One-Time Password (OTP) verification, and web technologies to provide secure file storage and management.

The proposed system protects sensitive files using Fernet-based symmetric encryption and allows authorized users to encrypt, decrypt, upload, download, and manage their files securely. JWT-based authentication is used to manage user sessions, while password hashing helps protect stored user credentials. Email-based OTP verification provides an additional layer of security for selected authentication and verification operations. The system also provides a secure vault for managing protected files and maintains activity records, including login activities, encryption and decryption operations, downloads, and other important events.

The proposed system aims to reduce the risks associated with unauthorized file access and improve the efficiency of secure digital file management. By combining encryption, authentication, verification, and activity logging, the system provides a unified platform for protecting sensitive files and managing user activities.

Keywords— Cybersecurity; File Encryption; Secure File Management; Fernet Symmetric Encryption; JWT Authentication; Password Hashing; Email OTP Verification; Secure File Storage; Activity Logging; Web Application Security; Data Confidentiality; Access Control.



I. INTRODUCTION

Digital technology has become an important part of everyday life, increasing the use of computers and web applications for storing, sharing, and managing digital files. Although digital storage provides convenience and accessibility, protecting sensitive information from unauthorized access, data theft, and misuse remains a major challenge. Confidential documents, academic records, and professional files may be exposed if appropriate security measures are not implemented. Traditional file management systems may not provide sufficient protection, making secure file handling an important requirement.

The proposed CipherCore Security Suite uses encryption, authentication, password hashing, and web technologies to improve file security. The system protects files using Fernet-based symmetric encryption and supports secure file encryption, decryption, storage, and management. JWT-based authentication manages user sessions, while email-based One-Time Password (OTP) verification provides an additional layer of security.

The application provides a secure vault that allows authorized users to manage protected files and includes activity logging to record important operations, such as logins, encryption, decryption, and downloads. Developed using React.js, Python Flask, and MySQL, the system integrates user interface management, backend processing, and database operations.

The main objective of the proposed system is to improve data confidentiality, strengthen access control, and simplify secure file management. By combining multiple security mechanisms within a single application, CipherCore aims to provide a convenient and organized environment for protecting sensitive digital information while supporting accountability through activity monitoring. Furthermore, the system provides a centralized platform for handling file-related operations and maintaining security records, helping users manage their digital information more effectively.

II. LITERATURE REVIEW

The protection of digital information has become an important research area as individuals and organizations increasingly depend on web applications to store, share, and manage confidential files. Previous research and established security standards have explored various techniques to protect sensitive information, including file encryption, password hashing, user authentication, and access control. The Advanced Encryption Standard (AES), published by the National Institute of Standards and Technology (NIST), provides a widely recognized method for protecting digital data through symmetric encryption. This approach allows information to be encrypted and decrypted using a secret key, making it useful for applications that require secure file storage. However, the effectiveness of encryption also depends on proper key management and access control.

Jones, Bradley, and Sakimura introduced the JSON Web Token (JWT) standard in RFC 7519 in 2015

[1]. JWT provides a compact format for transferring information between parties and is commonly used in web applications to support authentication and authorization. When implemented with appropriate signature verification, token expiration, and secure secret management, it helps applications manage access to protected resources. Similarly, password security plays an important role in protecting user accounts. NIST's Digital Identity Guidelines recommend appropriate password-hashing techniques to protect stored password credentials against unauthorized access

[2]. These security practices provide a foundation for developing web applications that require reliable user authentication and protection of confidential information.

Additional verification methods can provide another layer of protection for security-sensitive operations.



III. METHODOLOGY

The proposed CipherCore Security Suite follows a modular, security-focused approach that integrates file encryption, user authentication, email-based OTP verification, secure file storage, and activity monitoring. The system is designed to protect sensitive digital files, control access to authorized users, and maintain records of important file-related operations. It follows a client-server architecture in which the frontend provides the user interface, the backend handles authentication and file processing, and the database stores user information and activity records. The integration of these components provides a centralized platform for secure file management.

The methodology begins with user registration and authentication. Users create accounts by providing the required details, and their passwords are stored using password hashing techniques to reduce the risk of credential exposure. During login, the backend verifies the submitted credentials and generates a JSON Web Token (JWT) for authenticated access. Email-based One-Time Password (OTP) verification provides an additional verification step for selected security-sensitive operations. The system validates the OTP before allowing the corresponding operation to proceed, helping prevent unauthorized access to protected functions.

The file encryption and decryption process forms the core of CipherCore Security Suite. When a user selects a file for encryption, the backend processes the request and applies Fernet-based symmetric encryption to protect the file contents. The encrypted file is stored for subsequent access by authorized users. During decryption, the system validates the request and applies the corresponding decryption mechanism to recover the original file contents. The application also includes a secure vault that helps users manage their protected files. Appropriate validation and error handling are required to manage invalid requests, incorrect keys, and unsuccessful encryption or decryption attempts.

IV. RESULTS AND DISCUSSION

The proposed CipherCore Security Suite integrates file encryption and decryption, user authentication, email-based OTP verification, secure file storage, and activity logging within a single web application. The system is designed to protect confidential files, restrict access to authorized users, and maintain records of important security-related activities. Since verified experimental data are not yet available, numerical performance values are not included.

1. Homepage And Login Page

1.a Home Page

The homepage includes navigation options such as Features, Security, How It Works, About, Login, and Get Started, allowing users to explore the application and access their accounts. The interface highlights the main security functions, including AES-based encryption, email OTP verification, and a secure vault for protected files. These features are intended to make file protection more accessible while helping users manage confidential information in an organized manner.

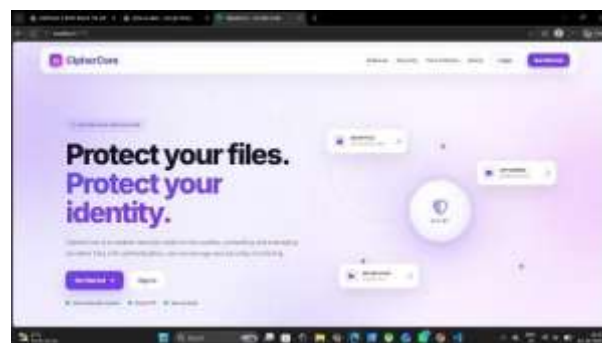


Fig 1.a Homepage.

1.b Login Page

The login page of CipherCore Security Suite provides a secure entry point for registered users to access the application's features. It includes email address and password fields, a password visibility option, a "Remember me" checkbox, and a password recovery link. New users can also navigate to the account creation page. The interface is designed to make the login process simple while providing access to protected files,



encryption tools, the secure vault, and account security controls. The authentication process verifies the user's credentials before granting access to authorized features.

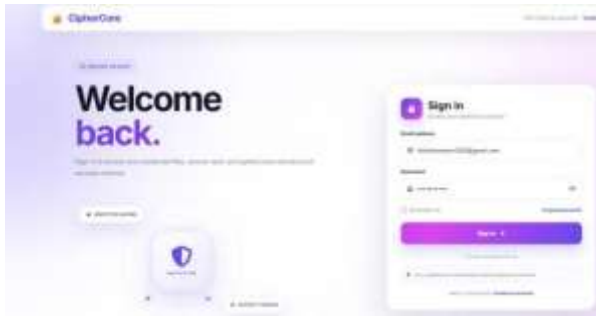


Fig 1.b Login Page

2 Dashboard

The dashboard of CipherCore Security Suite provides a centralized view of the user's account, file management activities, and security status. It displays key information, including the number of files stored in the secure vault, encrypted and decrypted files, and registered devices. The dashboard also provides quick access to important functions such as file encryption, decryption, vault management, and USB security.

The Recent Activity section displays security-related events, including file downloads and failed decryption attempts, helping users review their file operations. The sidebar provides navigation to the dashboard, encryption, decryption, vault, logs, and settings pages. A system status indicator gives users a quick overview of the account's displayed protection status. Overall, the dashboard is designed to simplify file management, improve visibility into security activities, and provide convenient access to the application's main features.

Overall, the CipherCore dashboard brings together file management, security status information, quick actions, and recent activity monitoring within a single interface. Its design supports The dashboard of CipherCore Security Suite provides a centralized view of the user's account, file management activities, and security status. It displays key information, including the number of files stored in the secure vault, encrypted and decrypted files, and registered devices. The dashboard also provides quick access to important functions such as file encryption, decryption, vault

management, and USB security.



Fig.2 Dashboard Page

3 Encryption Page

The file encryption interface is an important component of CipherCore Security Suite, designed to help users protect sensitive digital files from unauthorized access. The page provides a dedicated environment where users can select files for encryption and access the security options required to protect their information. Its simple layout allows users to understand the encryption process without navigating through multiple sections of the application.

The interface includes a file selection area that allows users to browse and choose a file for encryption. After selecting a file, users can provide the required encryption password and complete any additional verification steps implemented by the system. The selected file is then processed by the backend encryption service, which converts the original file contents into an encrypted format. The resulting encrypted file can be stored or downloaded according to the application's supported functionality.

The page also highlights encryption security through a dedicated security section and a visual indicator representing the encryption engine. These elements help users identify the purpose of the page and understand that the selected file is intended to be protected. The interface emphasizes the importance of using an appropriate encryption password and completing the required verification process before performing sensitive file operations.

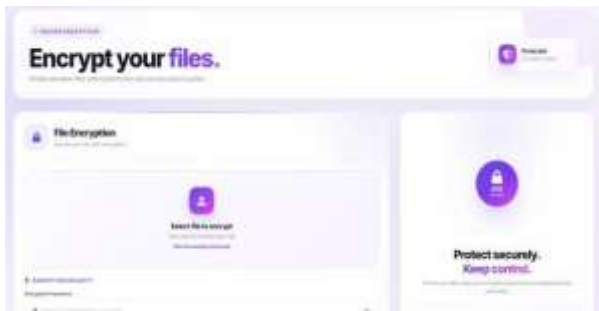


Fig.3 Encrypt Page

4. Decrypt Page

The file decryption interface of CipherCore Security Suite is designed to help authorized users recover their original files from encrypted versions. It provides a structured environment where users can select an encrypted file, enter the required password, and complete email-based One-Time Password (OTP) verification before proceeding with decryption. This process adds an additional verification step to help protect confidential files from unauthorized access. The interface displays the selected encrypted file along with its filename and size, allowing users to confirm which file they intend to decrypt. A decryption password field is provided, together with a password visibility option, so users can enter and verify their credentials. The Verify Password function checks the submitted password before the user proceeds to the email verification stage. This helps organize the decryption process into separate validation steps. The Email Verification OTP section allows users to request a six-digit verification code sent to their registered email address. After receiving the code, users can enter it into the designated fields and submit it for verification. The system is intended to permit decryption only after the required password and OTP checks have been completed successfully. Separating these steps provides an additional layer of protection for sensitive file operations.

From a technical perspective, the frontend collects the selected file, password, and OTP input, while the backend validates the request and performs the corresponding decryption operation. The application uses its configured encryption mechanism to restore the original file contents. Appropriate error handling is necessary to manage incorrect passwords, invalid or expired OTPs,

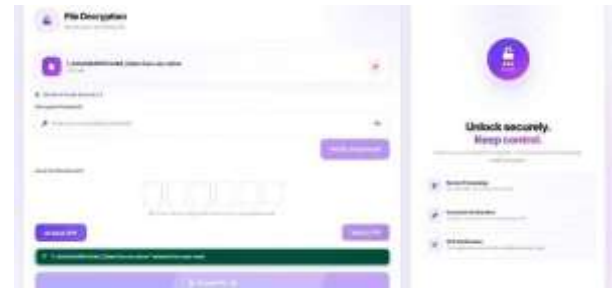


Fig 4. Decrypt Page

missing files, and unsuccessful decryption attempts. Successful operations and relevant failures can also be recorded in the activity logs to support security monitoring.

V. CONCLUSION

CipherCore Security Suite presents a practical approach to protecting and managing sensitive digital files through a web-based application. The system integrates file encryption and decryption, user authentication, password hashing, email-based One-Time Password (OTP) verification, secure file storage, and activity logging within a single platform. By bringing these features together, the application aims to make file protection more convenient while helping users maintain control over their confidential information. The use of React.js, Python Flask, and MySQL provides a structured foundation for the application's user interface, backend processing, and data management. The dashboard offers access to the main security functions, while the encryption and decryption modules provide dedicated interfaces for protecting and recovering files. The secure vault and activity logs further support file organization and visibility into security-related operations.

Although the system combines several established security mechanisms, its overall effectiveness depends on correct implementation, secure key management, reliable authentication, and appropriate access controls.

Further improvements may include enhanced key management, stronger multi-factor authentication, more detailed security monitoring, and comprehensive performance testing



In conclusion, CipherCore Security Suite demonstrates how multiple security mechanisms can be integrated into a single web application to support secure digital file management. With further testing and improvement, the system has the potential to provide a useful solution for users who need a convenient way to protect and manage confidential files.

VI. REFERENCES

- [1] National Institute of Standards and Technology, “Advanced Encryption Standard (AES),” Federal Information Processing Standards Publication 197, 2023.
- [2] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” Internet Engineering Task Force (IETF), RFC 7519, May 2015.
- [3] National Institute of Standards and Technology, “Digital Identity Guidelines: Authentication and Authenticator Management,” NIST Special Publication 800-63B-4, 2025.
- [4] OWASP Foundation, “Cryptographic Storage Cheat Sheet,” OWASP Cheat Sheet Series.
- [5] OWASP Foundation, “Password Storage Cheat Sheet,” OWASP Cheat Sheet Series.
- [6] The Python Cryptographic Authority, “Fernet (Symmetric Encryption),” Cryptography Documentation.
- [7] OWASP Foundation, “Key Management Cheat Sheet,” OWASP Cheat Sheet Series.