CHAPTER 4
RESULT AND FINDINGS

Introduction

The chapter demonstrates the findings that were acquired throughout the process of development and testing of Clouble: Secure Cloud Storage with Malware Analyzer and AES Encryption. The results are addressed in regard to the goals established during the previous stage of the project. This will involve study of the fundamental features of the system including malware scanning with VirusTotal API, the usage of AES to encrypt and decrypt, the generation of the QR keys, and the safe storage of files.

The result of the functional testing where the performance and security of the system are checked is also provided by this chapter. In each section, the behavior of the system under various conditions is emphasized, which will allow achieving the desired objectives in terms of protecting the data of users against malware and unauthorized access. This chapter also checks the overall performance of the system such as suitability and quick response.

4.1 Secure Cloud Storage with Malware Analyzer and AES Encryption

Clouble is a secure cloud storage system developed to address the rising concerns of malware infections and data breaches in cloud environments. It incorporates two key security features as used in malware analysis, a type of programmatic analysis using VirusTotal API and encryption of data through Advanced Encryption Standard (AES). This is because Clouble scans files with malware before they are uploaded to the cloud so that the files were clean and could be stored safely. When in case the file clears the scan, then AES is used to encrypt it after which the encrypted file is converted into a QR code thus allows easier handling and storage since it is in a visual format. Such dual layer mechanism adds shield to integrity and privacy of important files and this will be a better option than the traditional cloud storage platforms.

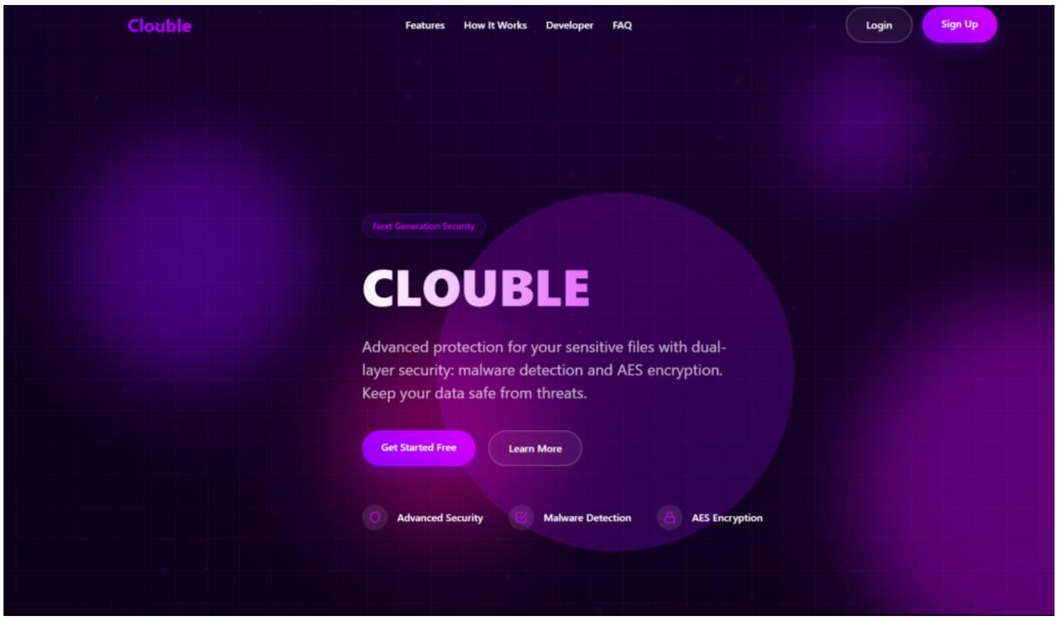
Other characteristics of the system are file management (upload, download, and view files, and delete files), and the interface of secure vault. Its interface is user-friendly, i.e., users can respond to encryption and scanning operations without having any knowledge about them. Using PHP, MySQL, and integration of API, Clouble provides a lightweight solution that can be scaled to meet an individual or an organizational need. In integrating the malware analysis and encryption technologies into a single platform, Clouble shows how the concept of proactive approach on security can be effectively deployed into a realistic file storage system.

4.2 User View

The user interface of the secure file storage system is designed to be intuitive and user-friendly, allowing users to manage their files securely using AES encryption. The following steps describe the user journey from registration to file management:

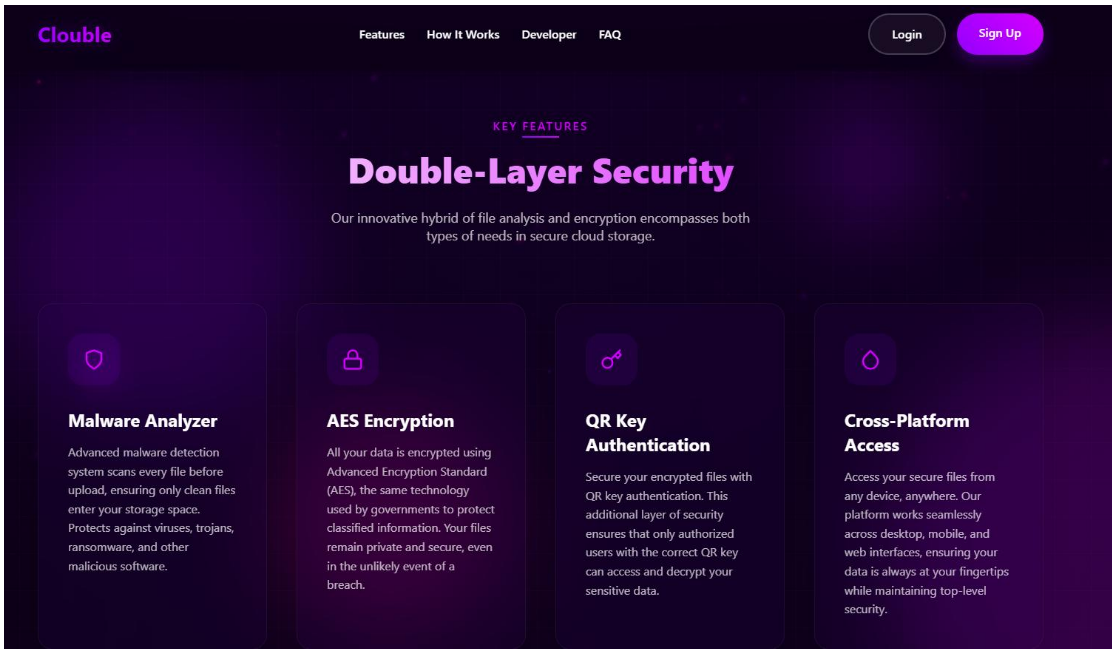
4.2.1 Access to Landing Page

The users will be redirected to the homepage whenever they enter the website



Advanced protection for your sensitive files with dual-layer security: malware detection and AES encryption. Keep your data safe from threats.

Figure 4. 1 Landing page view



Double-Layer Security

Our innovative hybrid of file analysis and encryption encompasses both types of needs in secure cloud storage.

Malware Analyzer

Advanced malware detection system scans every file before upload, ensuring only clean files enter your storage space. Protects against viruses, trojans, ransomware, and other malicious software.

AES Encryption

All your data is encrypted using Advanced Encryption Standard (AES), the same technology used by governments to protect classified information. Your files remain private and secure, even in the unlikely event of a breach.

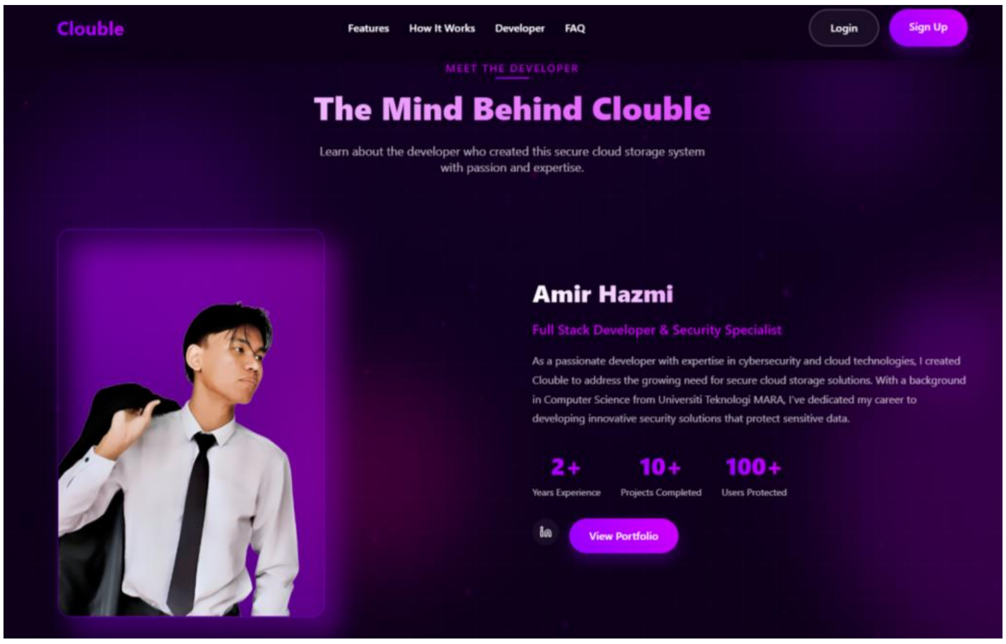
QR Key Authentication

Secure your encrypted files with QR key authentication. This additional layer of security ensures that only authorized users with the correct QR key can access and decrypt your sensitive data.

Cross-Platform Access

Access your secure files from any device, anywhere. Our platform works seamlessly across desktop, mobile, and web interfaces, ensuring your data is always at your fingertips while maintaining top-level security.

Figure 4. 2 Key Features



The Mind Behind Clouble

Learn about the developer who created this secure cloud storage system with passion and expertise.

Amir Hazmi

Full Stack Developer & Security Specialist

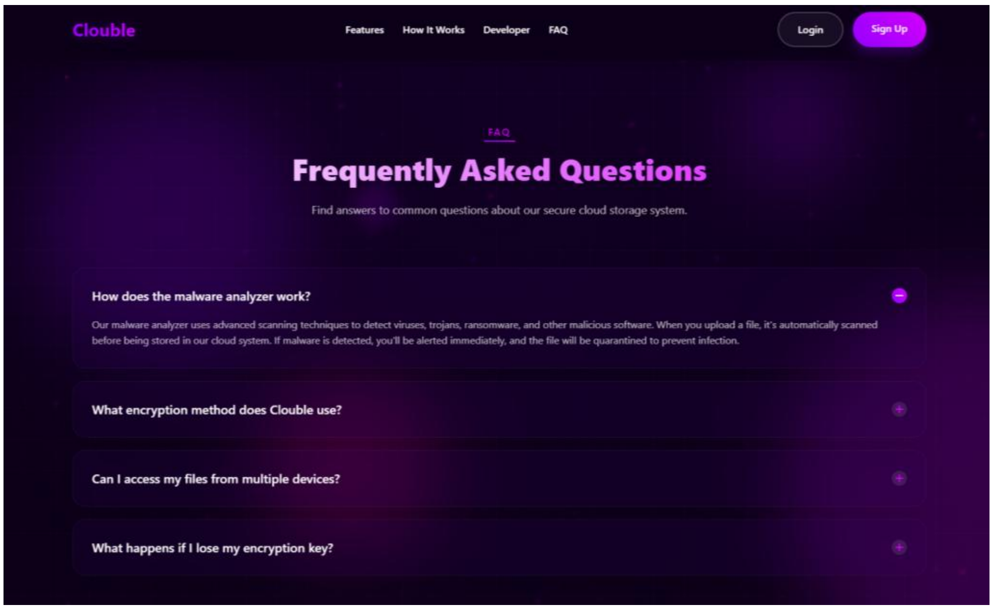
As a passionate developer with expertise in cybersecurity and cloud technologies, I created Clouble to address the growing need for secure cloud storage solutions. With a background in Computer Science from Universiti Teknologi MARA, I've dedicated my career to developing innovative security solutions that protect sensitive data.

2+ Years Experience

10+ Projects Completed

100+ Users Protected

Figure 4. 3 Developer Introduction



Frequently Asked Questions

Find answers to common questions about our secure cloud storage system.

How does the malware analyzer work?

Our malware analyzer uses advanced scanning techniques to detect viruses, trojans, ransomware, and other malicious software. When you upload a file, it's automatically scanned before being stored in our cloud system. If malware is detected, you'll be alerted immediately, and the file will be quarantined to prevent infection.

What encryption method does Clouble use?

Can I access my files from multiple devices?

What happens if I lose my encryption key?

Figure 4. 4 Frequently Asked Questions

The homepage view as shown at Figure 4.1 until Figure 4.4 which a landing page that will introduce to the user of overview of the system. This page also will redirect the user to login page or register page for the user to use the system.

4.2.2 User Registration

The user registers by providing their credentials, including username, first name, last name, phone number, password, and email address.

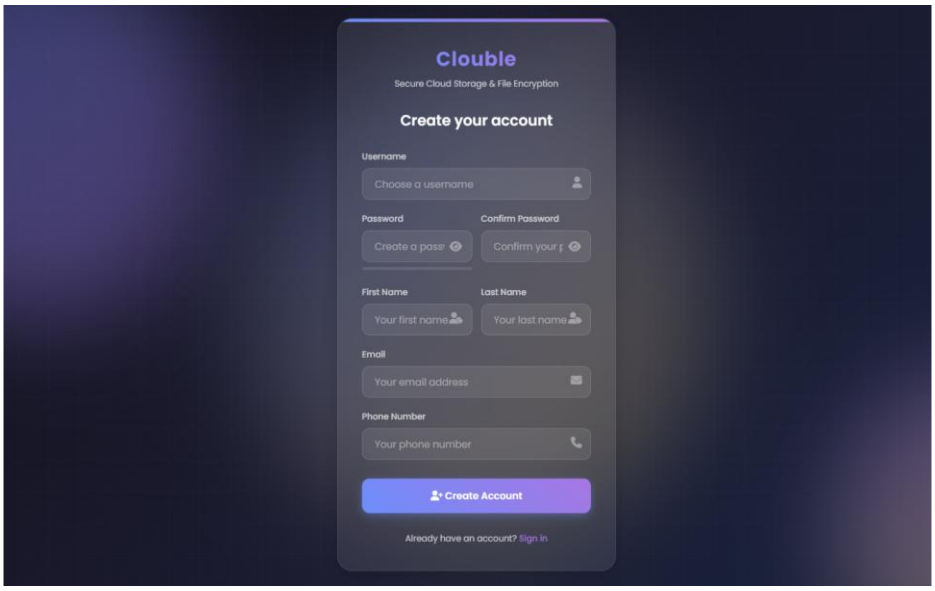
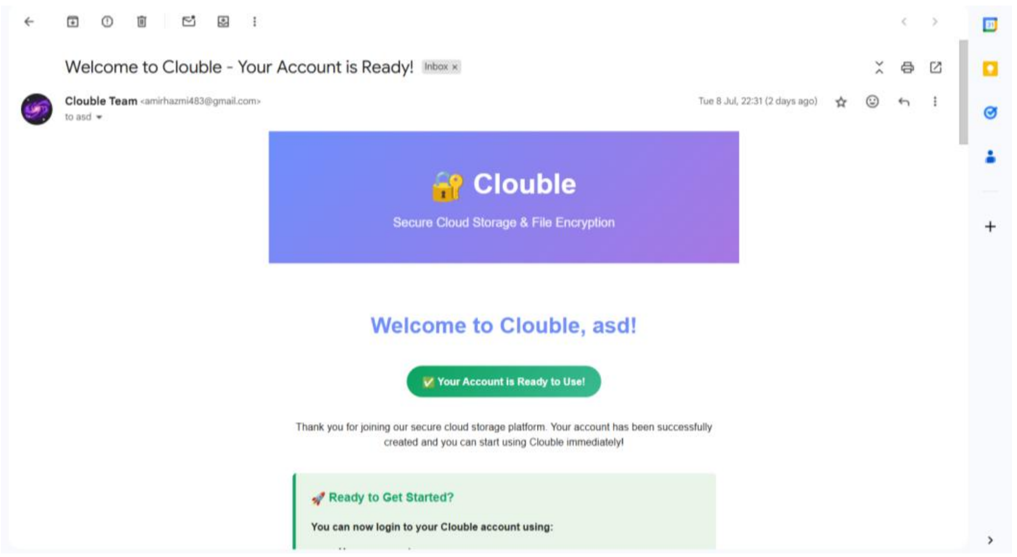


Figure 4. 5 Register page

The user registration screen, which allows new users to create accounts, is seen in Figure 4.5. Users must enter personal data, like their password, email address, and name. Before the user can access the system after submitting the registration form, the administrator must approve the account.

4.2.3 User Notification

Upon successful registration, the user receives an email confirming their registration.



Welcome to Clouble - Your Account is Ready! Inbox ×

Clouble Team <amirhazmi483@gmail.com>
to asd

Tue 8 Jul, 22:31 (2 days ago)

🔐 Clouble

Secure Cloud Storage & File Encryption

Welcome to Clouble, asd!

✅ Your Account is Ready to Use!

Thank you for joining our secure cloud storage platform. Your account has been successfully created and you can start using Clouble immediately!

🚀 Ready to Get Started?

You can now login to your Clouble account using:

Figure 4. 6 Registration Confirmation Email

The Figure 4.2 shows the confirmation email sent to users upon successful registration.

4.2.4 Access to Login Page

The user clicks the login link on the home page to access the secure environment.

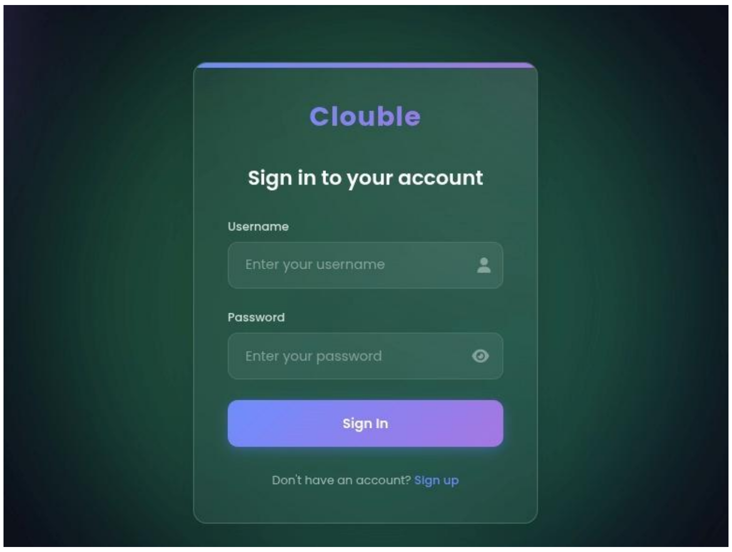
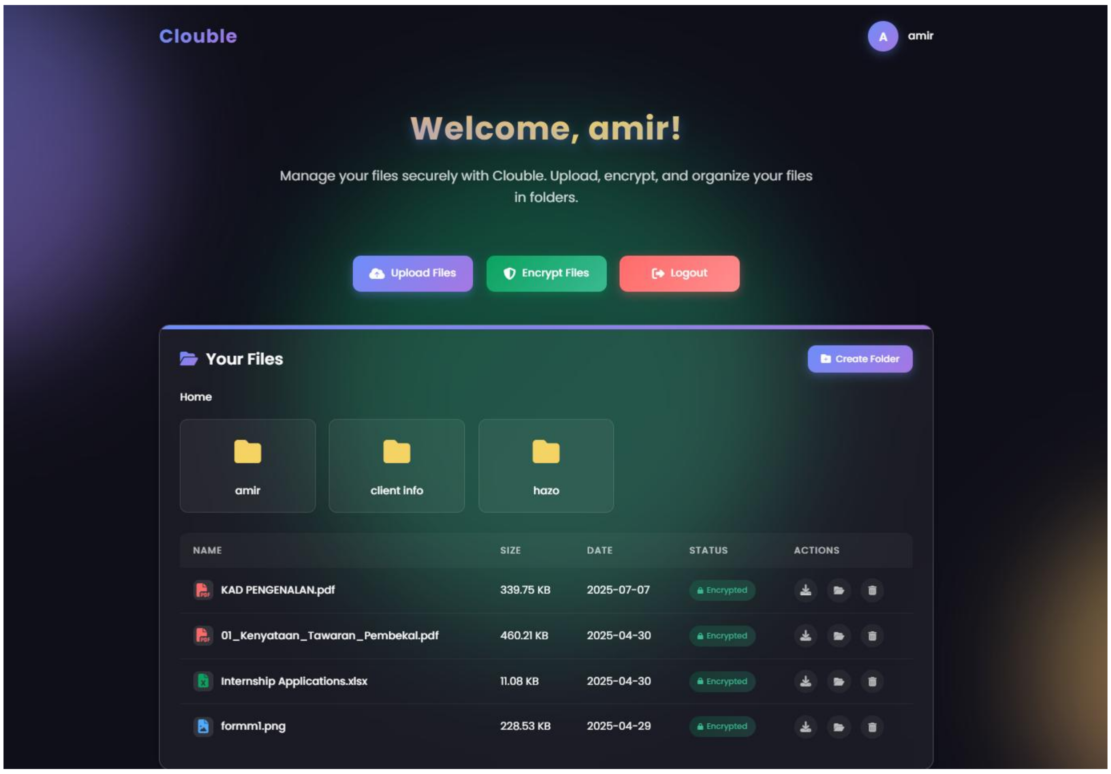


Figure 4. 7 Login Page

Figure 4.8 displays the home page of the secure file storage system. The page includes a login link that directs users to the login interface. It serves as the starting point for users to access their accounts.

4.2.5 Access to Dashboard Page

After successfully Registered and Logged in. User will redirect to the homepage of the system which shows the dashboard of the cloud storage



Welcome, amir!

Manage your files securely with Clouble. Upload, encrypt, and organize your files in folders.

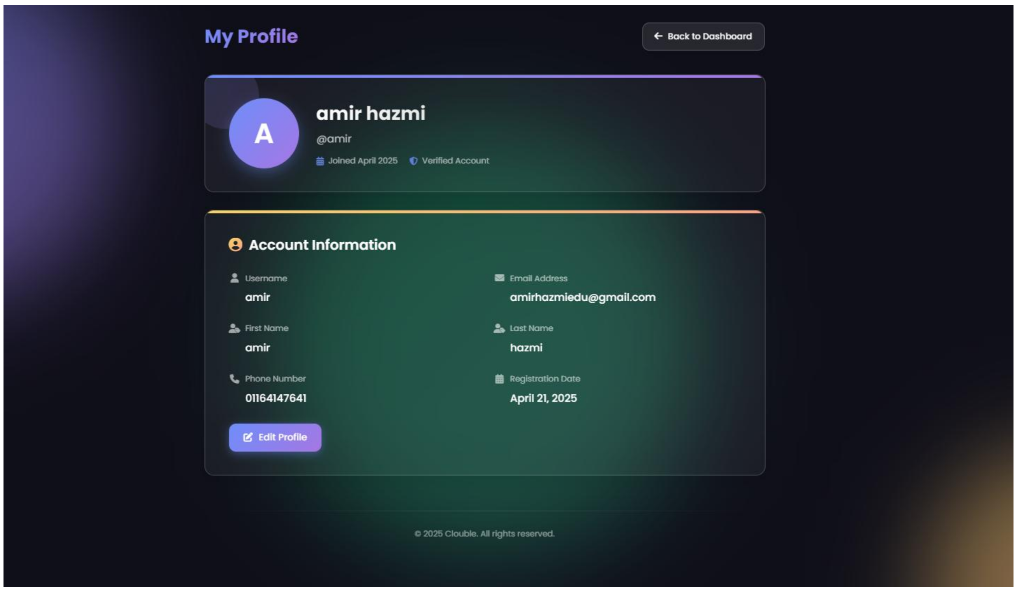
NAME	SIZE	DATE	STATUS	ACTIONS
KAD PENGENALAN.pdf	339.75 KB	2025-07-07	Encrypted	
01_Kenyataan_Tawaran_Pembekal.pdf	460.21 KB	2025-04-30	Encrypted	
Internship Applications.xlsx	11.08 KB	2025-04-30	Encrypted	
formm1.png	228.53 KB	2025-04-29	Encrypted	

Figure 4. 8:Cloud Home Interface

This Figure 4.8 shows the main interface that users see after logging in. The interface provides access to various features such as uploading, downloading, delete, view and encrypting, files. It acts as the user's control panel for managing their files securely.

4.2.6 Profile Management Page

The user can view their profile, update their information, and change their password if needed.



My Profile

Back to Dashboard

A

amir hazmi

@amir

Joined April 2025 · Verified Account

Account Information

Username	Email Address
amir	amirhazmiedu@gmail.com
First Name	Last Name
amir	hazmi
Phone Number	Registration Date
01164147641	April 21, 2025

Edit Profile

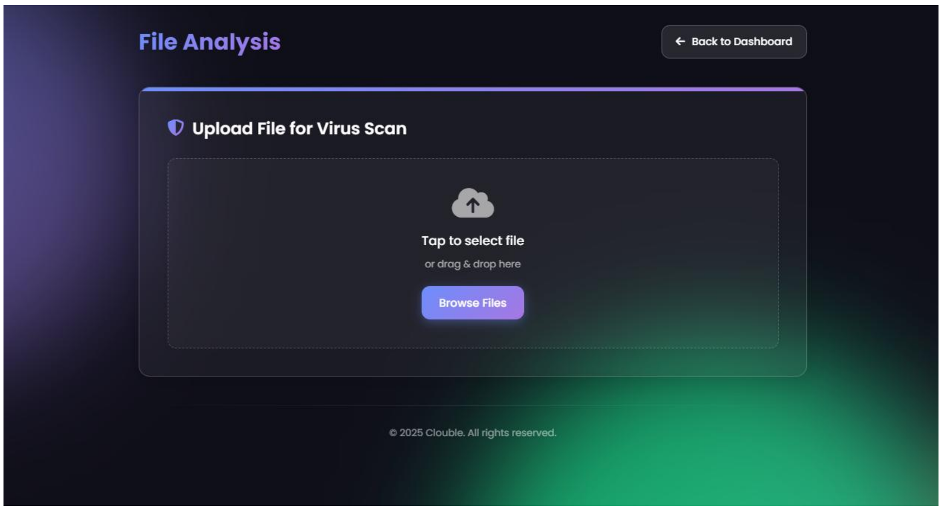
© 2025 Clouble. All rights reserved.

Figure 4. 9 Profile Management

Figure 4.9 illustrates the account settings page where users can view their personal information and change passwords. It ensures that users can maintain their account security and keep their information up to date.

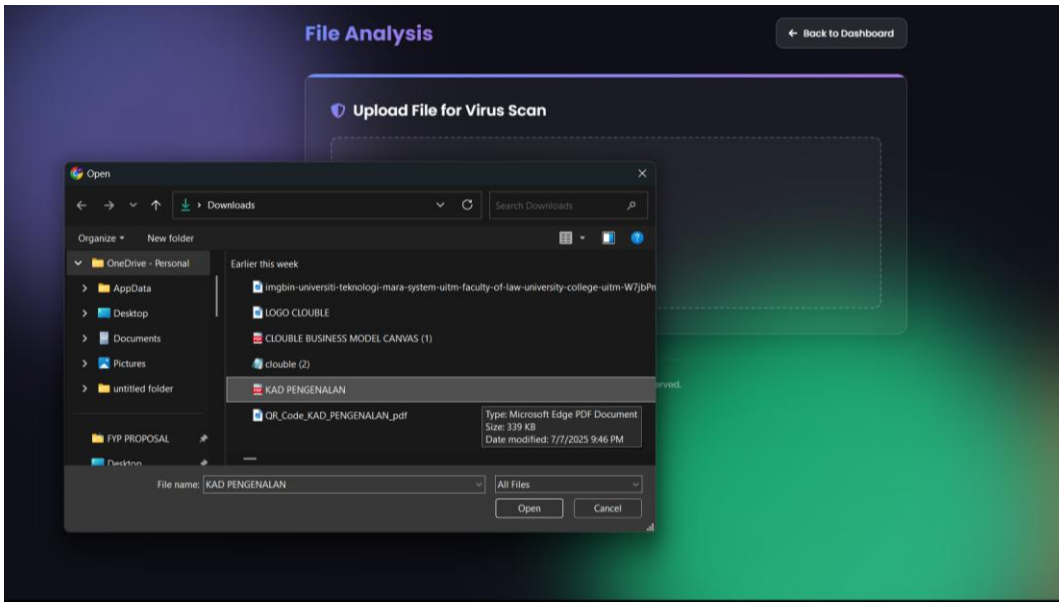
4.2.7 File Upload and Malware Analysis

On this page users are required to upload for the file to be analysed by a malware scanner integrated in the system using Virus Total API



.

Figure 4. 10 File Upload Interface

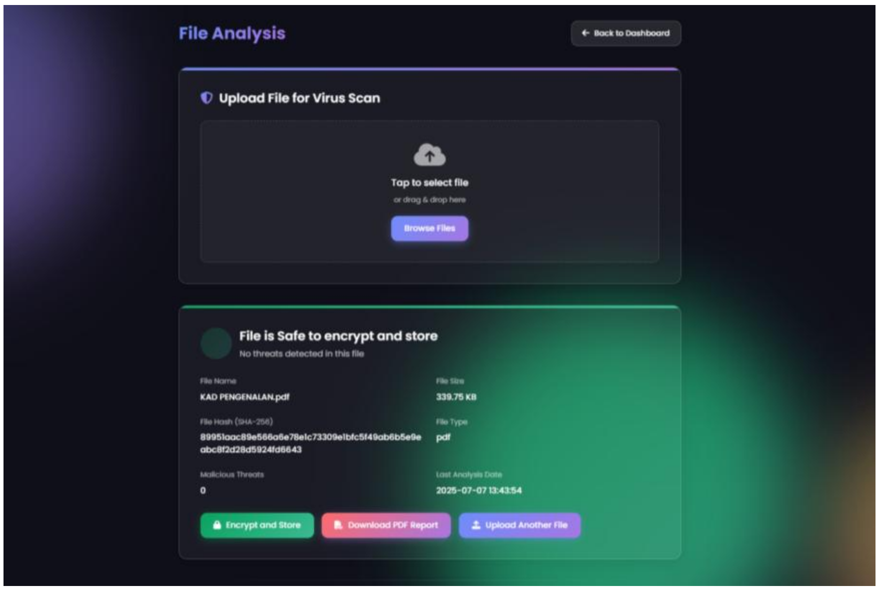


Type: Microsoft Edge PDF Document
Size: 339 KB
Date modified: 7/7/2025 9:46 PM

File name: KAD PENGENALAN

Figure 4. 11 User uploads file

Based on Figure 4.11, User then are required to select the file from their local device to be upload and analysed by malware scanning features on the system.



File is Safe to encrypt and store
No threats detected in this file

File Name
KAD PENGENALAN.pdf

File Size
339.75 KB

File Hash (SHA-256)
89951aac89e566a6e78e1c73309e1bfc5f49ab6b5e9e abc8f2d28d5924fd6643

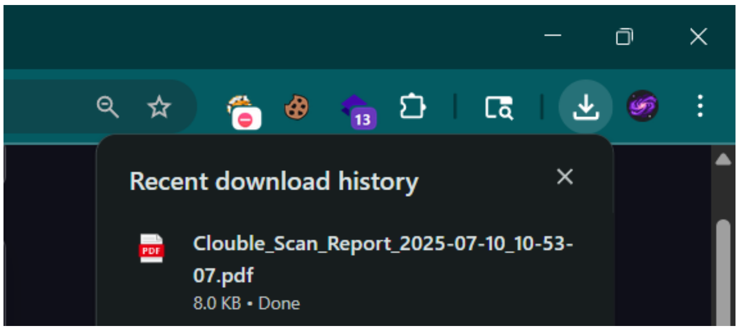
File Type
pdf

Malicious Threats
0

Last Analysis Date
2025-07-07 13:43:54

Figure 4. 12 File Scan result

On the figure 4.12, it shows that the file uploaded by the user are successfully scanned with a clean result. Proving that the files can be securely stored in the cloud storage system through the encryption. The results then will show in a table, listing the description of the file including file name, file hash, file size, file type, malicious threat and Last analysis date.

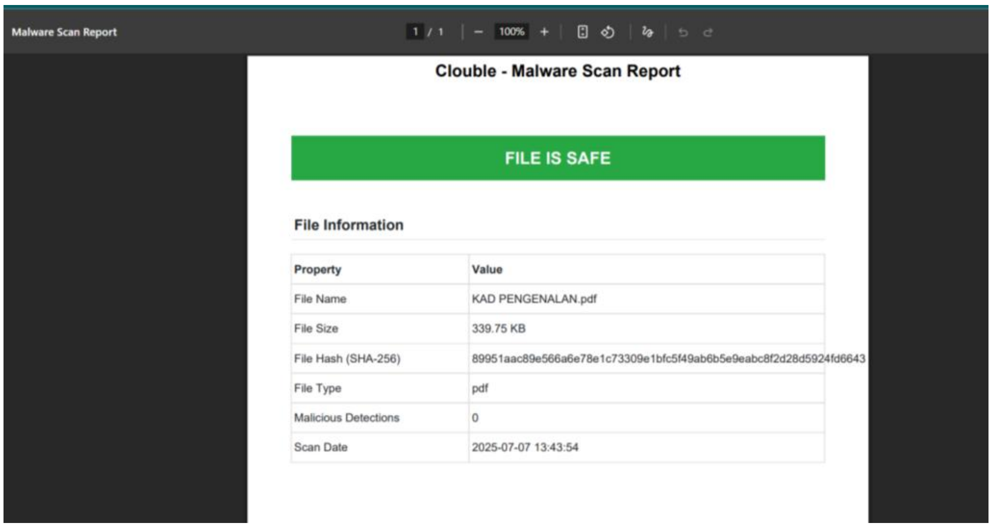


Recent download history

Clouble_Scan_Report_2025-07-10_10-53-07.pdf

8.0 KB • Done

Figure 4. 13 File scan report download



Clouble - Malware Scan Report

FILE IS SAFE

File Information

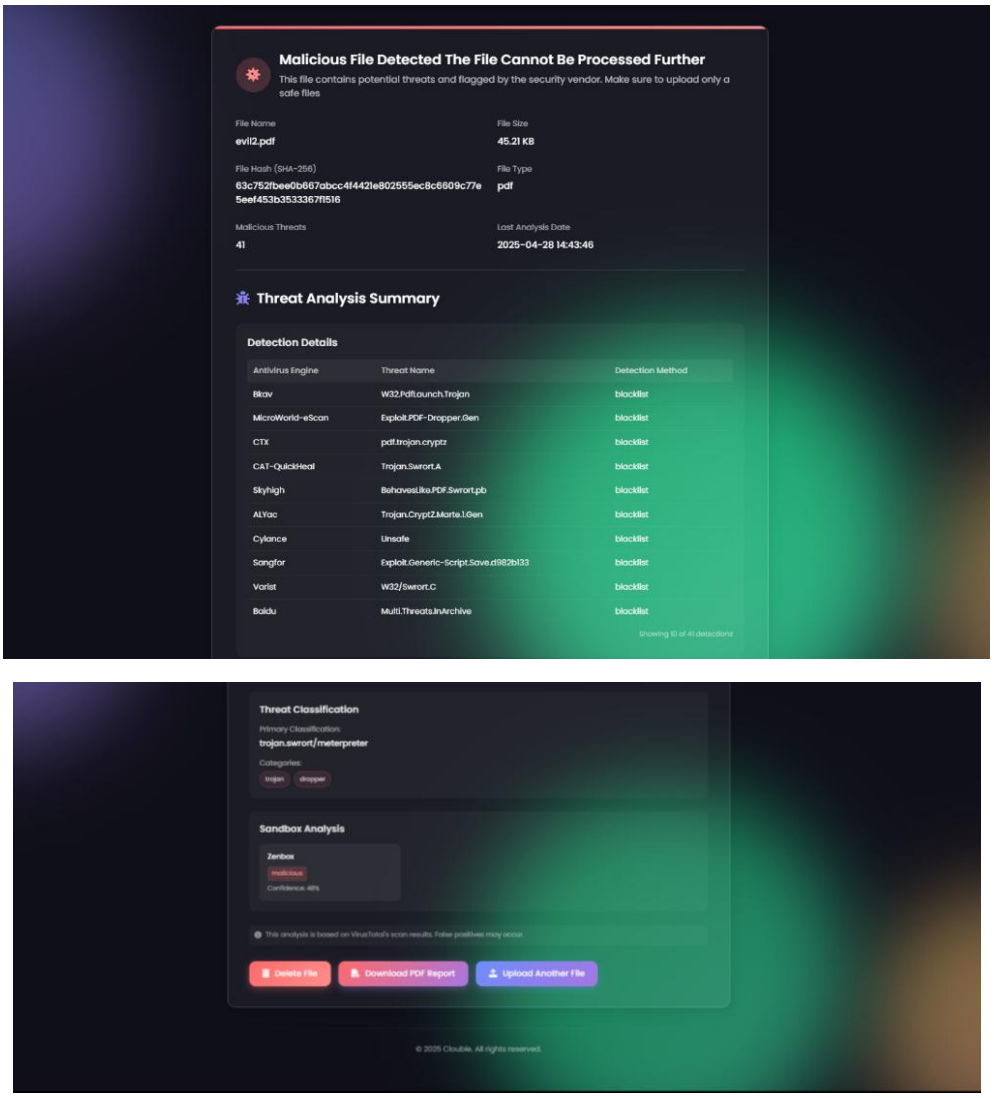
Property	Value
File Name	KAD PENGENALAN.pdf
File Size	339.75 KB
File Hash (SHA-256)	89951aac89e566a6e78e1c73309e1bfc5f49ab6b5e9eabc8f2d28d5924fd6643
File Type	pdf
Malicious Detections	0
Scan Date	2025-07-07 13:43:54

Figure 4. 14 File scan report

Based on figure 4.13 and 4.14. User than allowed to download the report result of the scanned file to view the report.

4.2.8 Flagged file and blacklisting

The results below shown the flagged files that are uploaded into the system



Malicious File Detected The File Cannot Be Processed Further

This file contains potential threats and flagged by the security vendor. Make sure to upload only a safe files

File Name evil2.pdf	File Size 45.21 KB
File Hash (SHA-256) 63c752fbee0b667abcc4f4421e802555ec8c6609c77e5eef453b3533367f1516	File Type pdf
Malicious Threats 41	Last Analysis Date 2025-04-28 14:43:46

Threat Analysis Summary

Detection Details

Antivirus Engine	Threat Name	Detection Method
Bkav	W32.PdfLaunch.Trojan	blacklist
MicroWorld-eScan	Exploit.PDF-Dropper.Gen	blacklist
CTX	pdf.trojan.cryptz	blacklist
CAT-QuickHeal	Trojan.Swrort.A	blacklist
Skyhigh	BehavesLike.PDF.Swrort.pb	blacklist
ALYac	Trojan.CryptZ.Marte.1.Gen	blacklist
Cylance	Unsafe	blacklist
Sangfor	Exploit.Generic-Script.Save.d982b133	blacklist
Varist	W32/Swrort.C	blacklist
Baidu	Multi.Threats.InArchive	blacklist

Showing 10 of 41 detections

Threat Classification

Primary Classification:
trojan.swrort/meterpreter

Categories:
trojan dropper

Sandbox Analysis

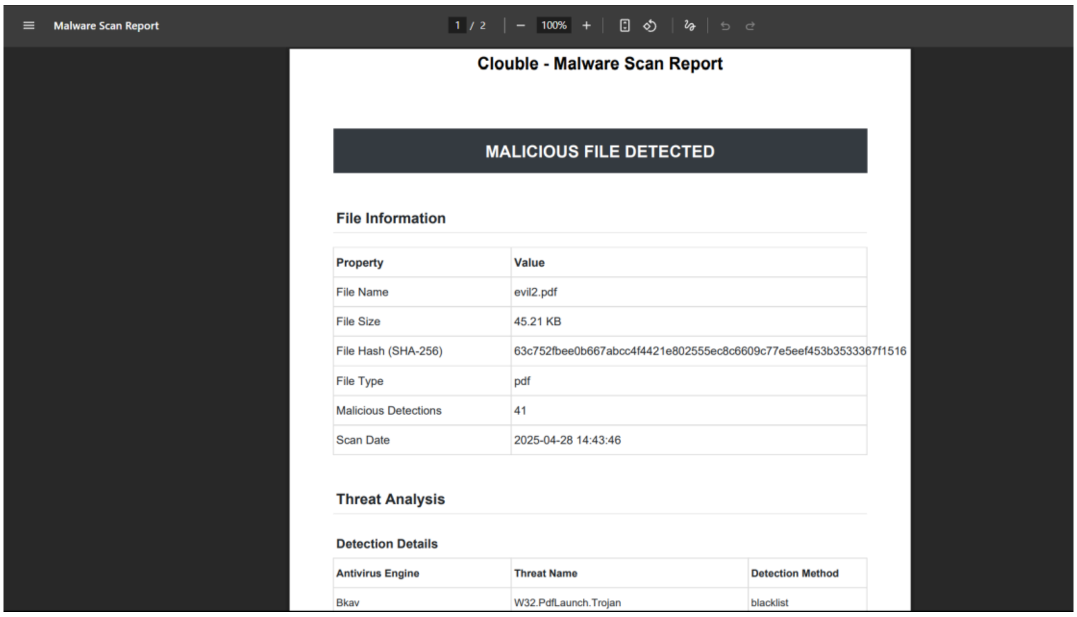
Zenbox
malicious
Confidence: 48%

This analysis is based on VirusTotal's scan results. False positives may occur.

© 2025 Cloudie. All rights reserved.

Figure 4. 15 Flagged File result

Figure 4.15 shows the result of flagged file, that contains a malware embedded program in the file. Flagged file then cannot proceed with encryption and file storing in the system.



Clouble - Malware Scan Report

MALICIOUS FILE DETECTED

File Information

Property	Value
File Name	evil2.pdf
File Size	45.21 KB
File Hash (SHA-256)	63c752fbee0b667abcc4f4421e802555ec8c6609c77e5eef453b3533367f1516
File Type	pdf
Malicious Detections	41
Scan Date	2025-04-28 14:43:46

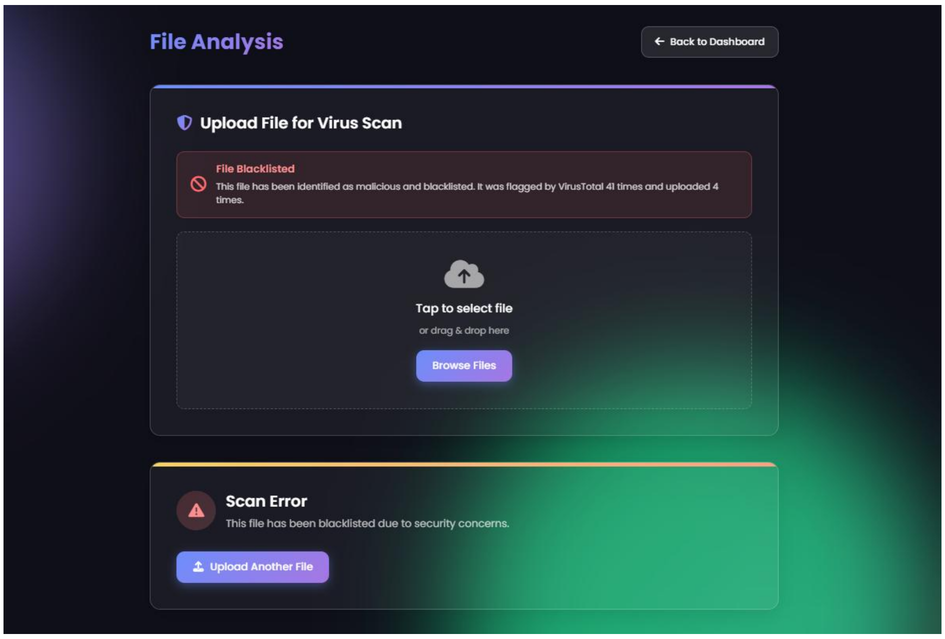
Threat Analysis

Detection Details

Antivirus Engine	Threat Name	Detection Method
Bkav	W32.PdfLaunch.Trojan	blacklist

Figure 4. 16 Malicious File report

Based on figure 4.16, User than allowed to download the report result of the scanned file to view the report. Including threat analysis and threat classification.



File Blacklisted

This file has been identified as malicious and blacklisted. It was flagged by VirusTotal 41 times and uploaded 4 times.

Scan Error

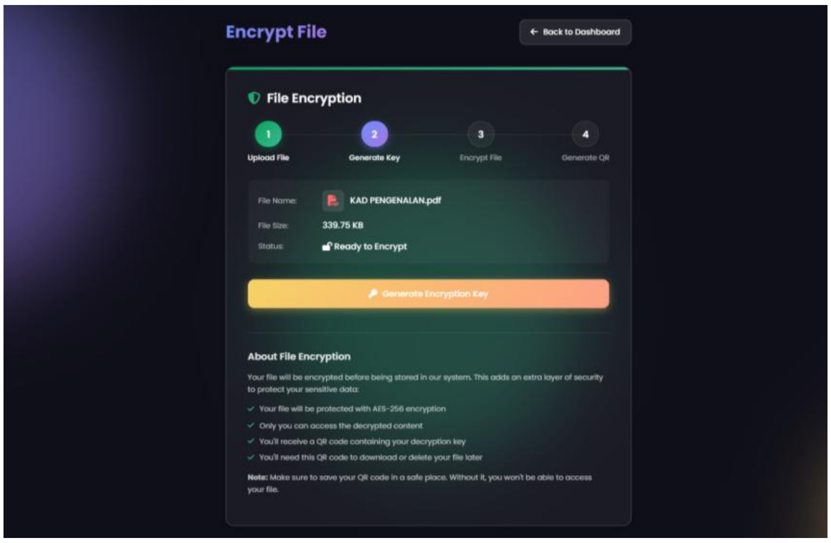
This file has been blacklisted due to security concerns.

Figure 4. 17 Blacklisting Malicious file

After multiple attempts of uploading same malicious file, the system then will blacklist the file from upload to be processed on the system.

4.2.9 File Encryption

After successful scanning of safe file, user can proceed to perform file encryption using AES key before storingthe file.



File Name:	KAD PENGENALAN.pdf
File Size:	339.75 KB
Status:	Ready to Encrypt

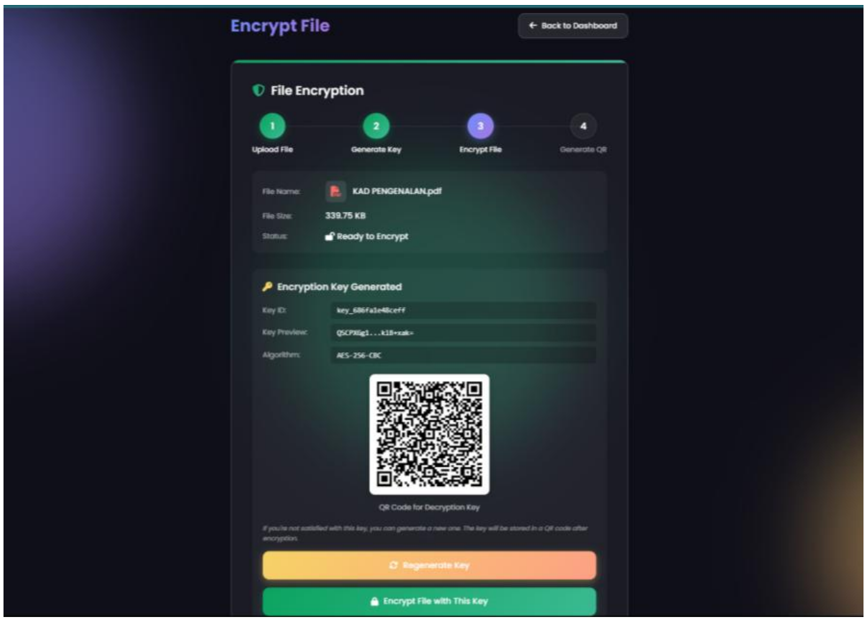
About File Encryption

Your file will be encrypted before being stored in our system. This adds an extra layer of security to protect your sensitive data:

- ✓ Your file will be protected with AES-256 encryption
- ✓ Only you can access the decrypted content
- ✓ You'll receive a QR code containing your decryption key
- ✓ You'll need this QR code to download or delete your file later

Note: Make sure to save your QR code in a safe place. Without it, you won't be able to access your file.

Figure 4. 18 File ready for encryption



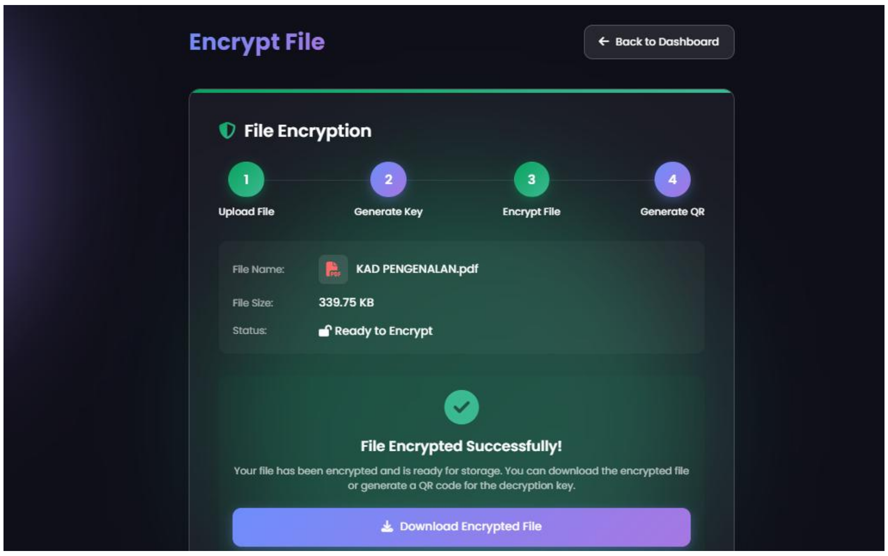
File Name:	KAD PENGENALAN.pdf
File Size:	339.75 KB
Status:	Ready to Encrypt

Encryption Key Generated

Key ID:	key_686fa1e48ceff
Key Preview:	QSCPXGg1...k18+xak=
Algorithm:	AES-256-CBC

If you're not satisfied with this key, you can generate a new one. The key will be stored in a QR code after encryption.

Figure 4. 19 File Encryption


Encrypt File

File Encryption

File Name: KAD PENGENALAN.pdf

File Size: 339.75 KB

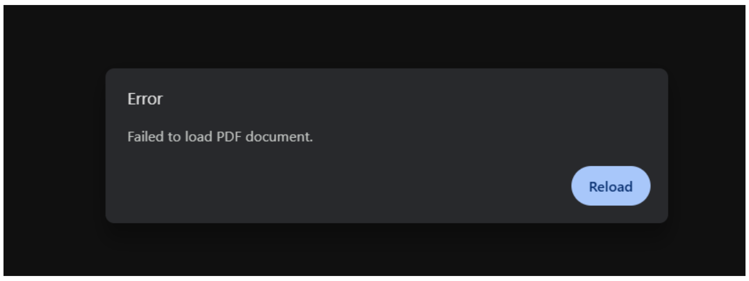
Status: Ready to Encrypt

File Encrypted Successfully!

Your file has been encrypted and is ready for storage. You can download the encrypted file or generate a QR code for the decryption key.

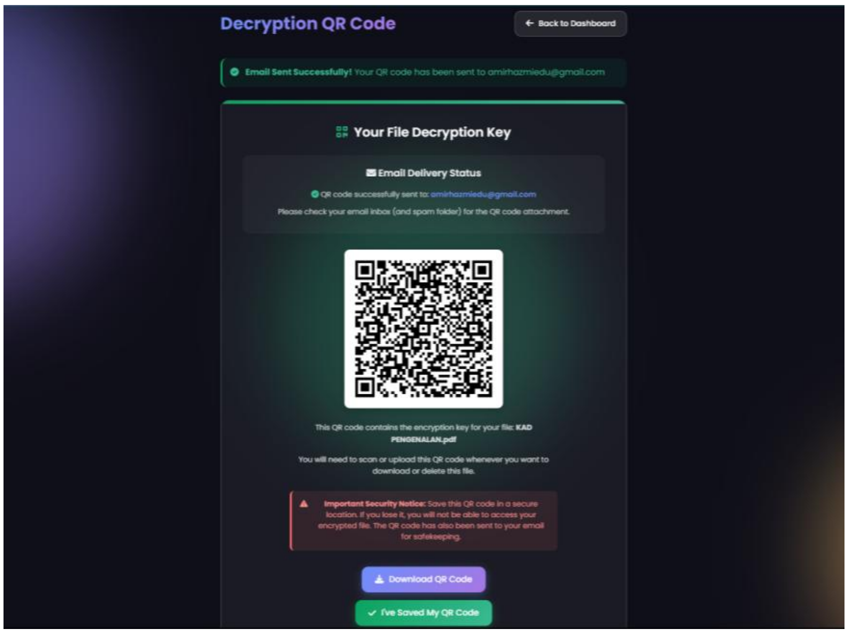
Figure 4. 20 Successful encryption

Based on Figure 4.19 and 4.20 The system generates an AES key and perform an encryption to the file


Error

Failed to load PDF document.

Figure 4. 21 Encrypted file


Decryption QR Code

Email Sent Successfully! Your QR code has been sent to amirhazmiedu@gmail.com

Your File Decryption Key

Email Delivery Status

QR code successfully sent to: amirhazmiedu@gmail.com

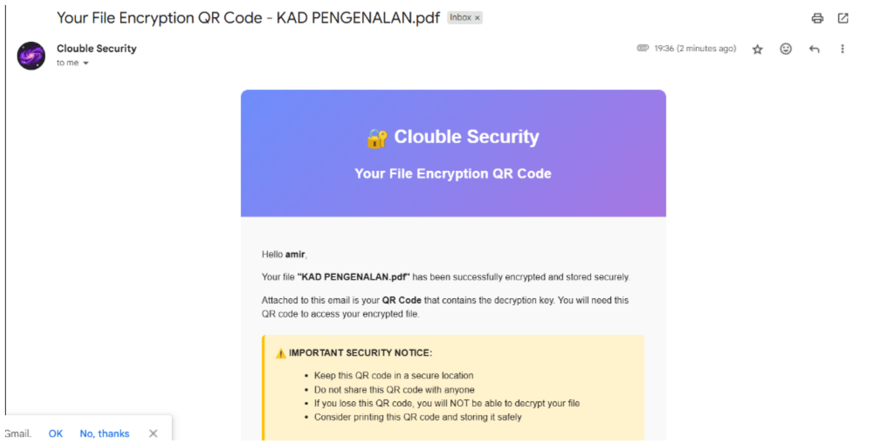
Please check your email inbox (and spam folder) for the QR code attachment.

This QR code contains the encryption key for your file: **KAD PENGENALAN.pdf**

You will need to scan or upload this QR code whenever you want to download or delete this file.

Important Security Notice: Save this QR code in a secure location. If you lose it, you will not be able to access your encrypted file. The QR code has also been sent to your email for safekeeping.

Figure 4. 22 QR Code key Download


Your File Encryption QR Code - KAD PENGENALAN.pdf Inbox ×

Clouble Security
to me

19:36 (2 minutes ago)

Clouble Security

Your File Encryption QR Code

Hello **amir**,

Your file **"KAD PENGENALAN.pdf"** has been successfully encrypted and stored securely.

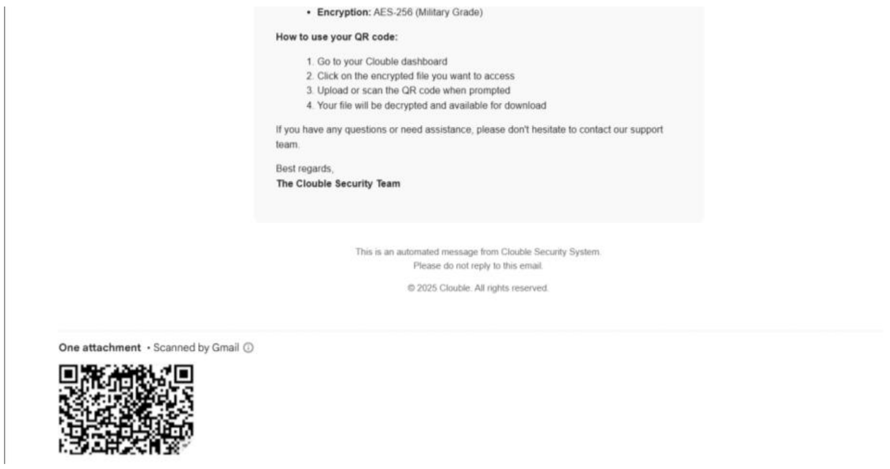
Attached to this email is your **QR Code** that contains the decryption key. You will need this QR code to access your encrypted file.

IMPORTANT SECURITY NOTICE:

- Keep this QR code in a secure location
- Do not share this QR code with anyone
- If you lose this QR code, you will NOT be able to decrypt your file
- Consider printing this QR code and storing it safely

Gmail. OK No, thanks

Figure 4. 23 QR Code Email


- **Encryption:** AES-256 (Military Grade)

How to use your QR code:

1. Go to your Clouble dashboard
2. Click on the encrypted file you want to access
3. Upload or scan the QR code when prompted
4. Your file will be decrypted and available for download

If you have any questions or need assistance, please don't hesitate to contact our support team.

Best regards,
The Clouble Security Team

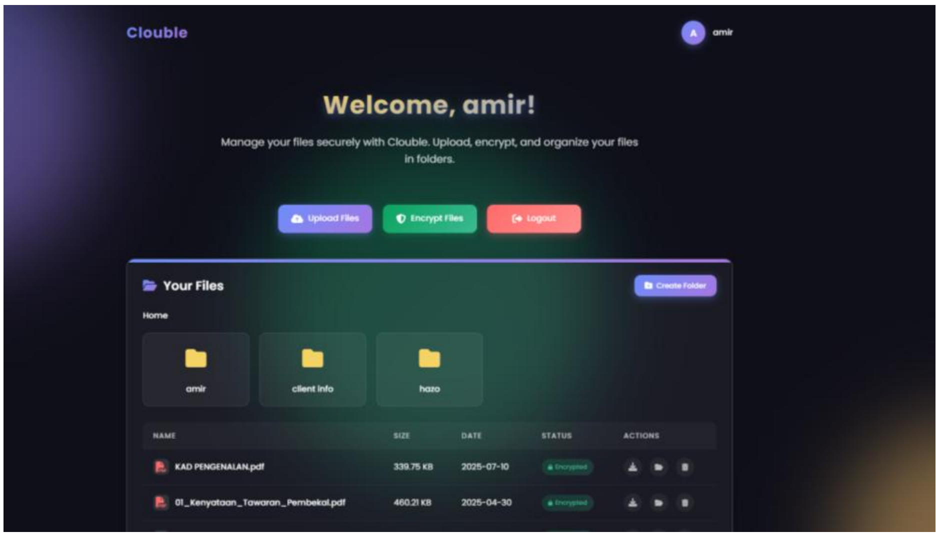
This is an automated message from Clouble Security System.
Please do not reply to this email.

© 2025 Clouble. All rights reserved.

One attachment • Scanned by Gmail

Figure 4. 24 QR Code Image attached

Based on Figure 4.23 until 4.24. The system then sends a QR Code Decryption key to the email .It provides an option to download the key as a QR code for easy storage.



Manage your files securely with Clouble. Upload, encrypt, and organize your files in folders.

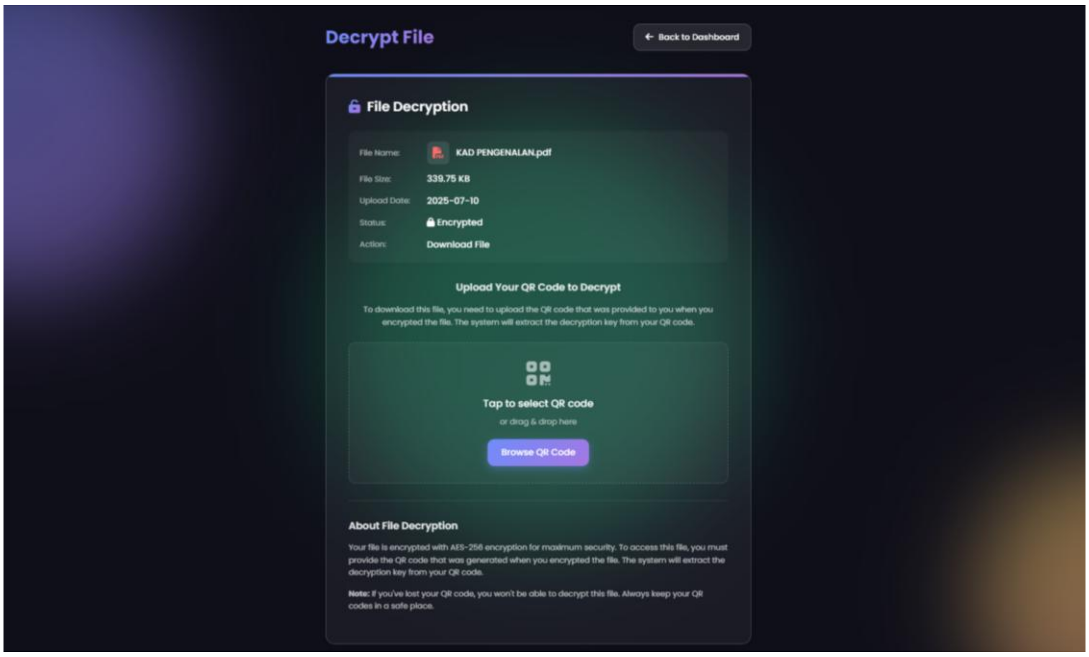
NAME	SIZE	DATE	STATUS	ACTIONS
KAD PENGENALAN.pdf	339.75 KB	2025-07-10	Encrypted	
01_Kenyataan_Tawaran_Pembekal.pdf	460.21 KB	2025-04-30	Encrypted	

Figure 4. 25 Stored file in Dashboard Page

Encrypted files are now stored and displayed on the dashboard page of the user also the dashboard will able the user to create the folder and arrange the encrypted files on the dashboard.

4.2.10 File Decryption

To decrypt a file, the user need to uploads the QR code containing the key. If the key is correct, the file is decrypted and available for any actions further.



File Name:	KAD PENGENALAN.pdf
File Size:	339.75 KB
Upload Date:	2025-07-10
Status:	Encrypted
Action:	Download File

Upload Your QR Code to Decrypt

To download this file, you need to upload the QR code that was provided to you when you encrypted the file. The system will extract the decryption key from your QR code.

About File Decryption

Your file is encrypted with AES-256 encryption for maximum security. To access this file, you must provide the QR code that was generated when you encrypted the file. The system will extract the decryption key from your QR code.

Note: If you've lost your QR code, you won't be able to decrypt this file. Always keep your QR codes in a safe place.

Figure 4. 26 File Decryption Interface

Figure 4.26 illustrates the interface for decrypting files. Users need to provide the AES key to decrypt a file, ensuring that only authorized users can access the content of encrypted files.

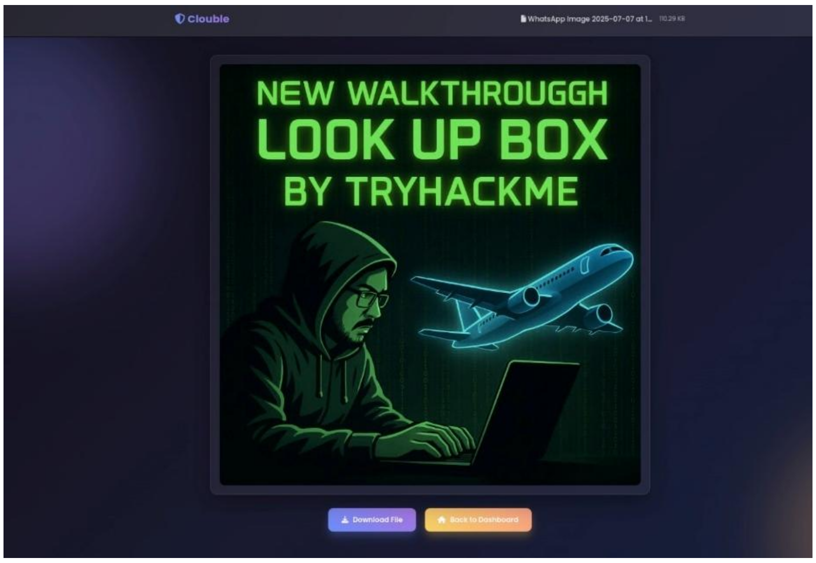
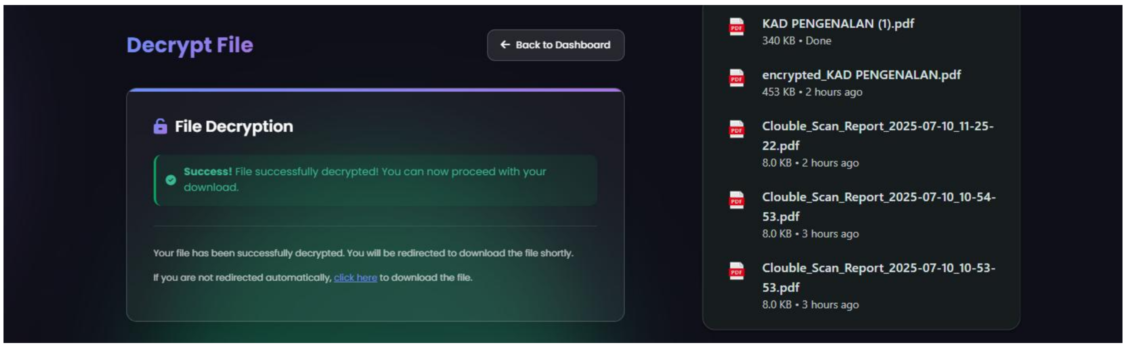


Figure 4. 27 File view after decryption

Figure 4.27 shows the blob file format used for viewing files after decryption. The system converts decrypted files into a blob format for secure viewing within the browser, maintaining the integrity of the file.

4.2.11 Download and Delete Files

The user can also download and delete files by providing the correct decryption key or QR code.



Success! File successfully decrypted! You can now proceed with your download.

Your file has been successfully decrypted. You will be redirected to download the file shortly.

If you are not redirected automatically, click here to download the file.

KAD PENGENALAN (1).pdf
340 KB • Done

encrypted_KAD PENGENALAN.pdf
453 KB • 2 hours ago

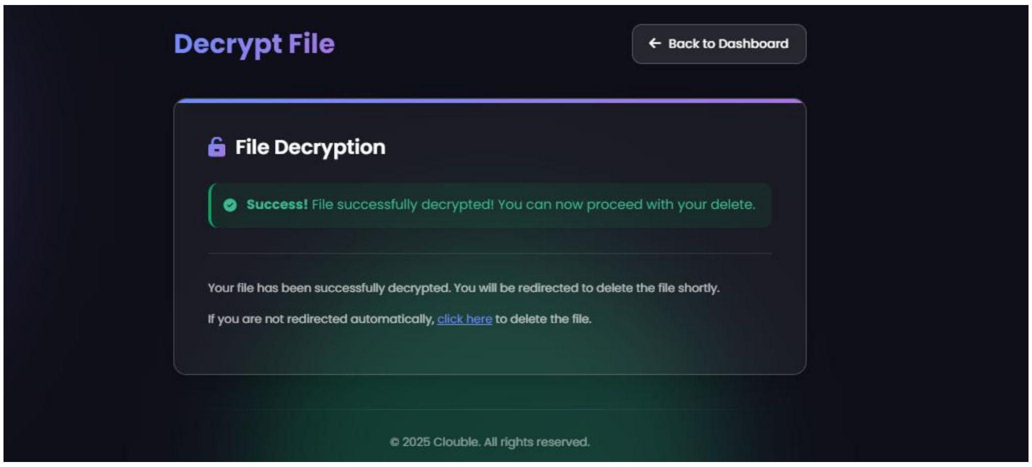
Clouble_Scan_Report_2025-07-10_11-25-22.pdf
8.0 KB • 2 hours ago

Clouble_Scan_Report_2025-07-10_10-54-53.pdf
8.0 KB • 3 hours ago

Clouble_Scan_Report_2025-07-10_10-53-53.pdf
8.0 KB • 3 hours ago

Figure 4. 28 File Download interface

Figure 4.28 depicts the interface for downloading files. Users can select files from their storage and download them to their local device. The system ensures that files are securely downloaded without compromising their integrity.



Success! File successfully decrypted! You can now proceed with your delete.

Your file has been successfully decrypted. You will be redirected to delete the file shortly.

If you are not redirected automatically, click here to delete the file.

© 2025 Clouble. All rights reserved.

Figure 4. 29 File Delete Interface

The interface for removing files from the system is shown in Figure 4.29. To ensure that undesired data are deleted from the storage, users can choose the items they no longer require and securely delete them.

4.3 Testing Results

The testing process was divided into several categories to thoroughly evaluate the system's performance:

1. Functional testing
2. Encryption and Decryption Performance
3. Storage of Encrypted Files on the Server
4. Data Integrity Verification
5. Malware Scanning and analysis

4.3.1 Functional testing

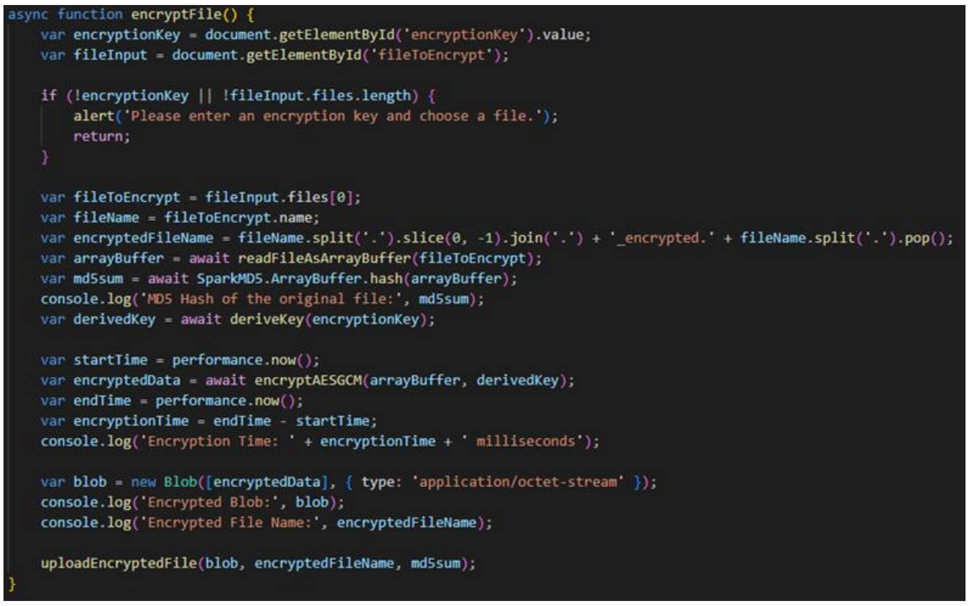
The secure cloud storage system underwent comprehensive testing to evaluate its functionality, accuracy, and usability. The testing process follows industry-standard practices, encompassing both positive and negative scenarios to assess the system's response under different conditions. Table 4.1 shows the functionality testing output for the system.

Table 4. 1:Functionality Testing Output

Test ID	Test Case Description	Expected Output	Result
1.	User Registration	Success	Pass
2.	User Login	Success	Pass
3.	File Upload	Success	Pass
4.	File Encryption	Success	Pass
5.	File Decryption	Success	Pass
6.	File Download	Success	Pass
7.	File Deletion	Success	Pass
8.	Malware Scanning	Success	Pass
9.	Malware Analyzing	Success	Pass
10.	Blacklisting Malicious File	Success	Pass

4.3.2 Encryption and Decryption Performance

In this section, this report will evaluate the performance of the AES encryption and decryption processes for various file types and sizes. Figure 4.31 shows the code snippet used to measure the encryption and decryption times in JavaScript. To conduct this evaluation, The system selected files of different sizes, specifically 1KB, 100KB, 1MB, and 10MB, and of different types, including .txt, .pdf, .jpeg, and .docx. The performance measurement involved several key steps. First, each file was read, and its content was converted into an ArrayBuffer using the FileReader API, ensuring asynchronous and efficient reading of the file. Next, we derived the AES encryption key from a user-provided password utilizing the PBKDF2 (Password-Based Key Derivation Function 2) algorithm. This process involved encoding the password into a Uint8Array with Text Encoder, importing it as raw key material, and then deriving the AES key with specified parameters for salt, iteration count, and hash function (SHA-256).



```javascript
async function encryptFile() {
    var encryptionKey = document.getElementById('encryptionKey').value;
    var fileInput = document.getElementById('fileToEncrypt');

    if (!encryptionKey || !fileInput.files.length) {
        alert('Please enter an encryption key and choose a file.');
        return;
    }

    var fileToEncrypt = fileInput.files[0];
    var fileName = fileToEncrypt.name;
    var encryptedFileName = fileName.split('.').slice(0, -1).join('.') + '_encrypted.' + fileName.split('.').pop();
    var arrayBuffer = await readFileAsArrayBuffer(fileToEncrypt);
    var md5sum = await SparkMD5.ArrayBuffer.hash(arrayBuffer);
    console.log('MD5 Hash of the original file:', md5sum);
    var derivedKey = await deriveKey(encryptionKey);

    var startTime = performance.now();
    var encryptedData = await encryptAESGCM(arrayBuffer, derivedKey);
    var endTime = performance.now();
    var encryptionTime = endTime - startTime;
    console.log('Encryption Time: ' + encryptionTime + ' milliseconds');

    var blob = new Blob([encryptedData], { type: 'application/octet-stream' });
    console.log('Encrypted Blob:', blob);
    console.log('Encrypted File Name:', encryptedFileName);

    uploadEncryptedFile(blob, encryptedFileName, md5sum);
}
```

Figure 4. 30 Code Snippet to Measure Time in JavaScript

Once the key was derived, the encryption of the file content using the AES-GCM (Galois/Counter Mode) encryption algorithm was carried out. During encryption, a random initialization vector (IV) was generated, and the data was encrypted with the AES key and IV. The encrypted data, combined with the IV, was then prepared as a single array for secure storage. To measure the time taken for these operations, the performance.now () method in JavaScript was used, capturing the precise start and end times for both encryption and decryption processes

Table 4. 2:AES Encryption and Decryption Time

File type	File Size	AES Encryption Time (ms)	AES Decryption Time (ms)
.txt	1KB	0.5	0.4
.pdf	100KB	0.6	0.5
.jpeg/png	1MB	0.7	0.6
.docx	10MB	1.0	0.9

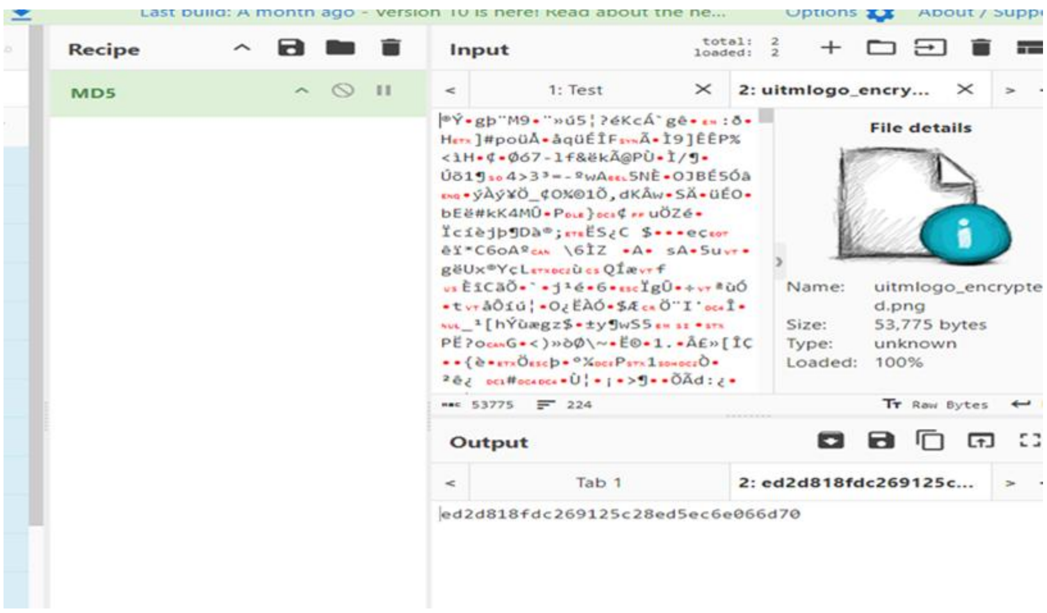
Table 4.2 presents the encryption and decryption times for these files, showing that the time required for encryption increases with file size. Smaller files like a 1KB .txt file are encrypted more quickly compared to larger files like a 10MB .docx file. The decryption times are slightly lower than the encryption times, indicating efficient decryption performance. The analysis shows that file content

type does not significantly impact encryption and decryption times, as files of the same size but different types exhibited similar performance metrics. This comprehensive testing approach demonstrates the robustness and efficiency of the AES encryption implementation in the secure file storage system.

The performance analysis of the AES encryption and decryption processes reveals several important insights. The results indicate that the time required to encrypt files increases proportionally with the file size. For instance, smaller files, such as a 1KB .txt file, are encrypted much more quickly compared to larger files like a 10MB .docx file. This is expected, as the encryption algorithm processes more data when dealing with larger files, leading to longer encryption times.

4.3.3 Storage of Encrypted Files on the Server

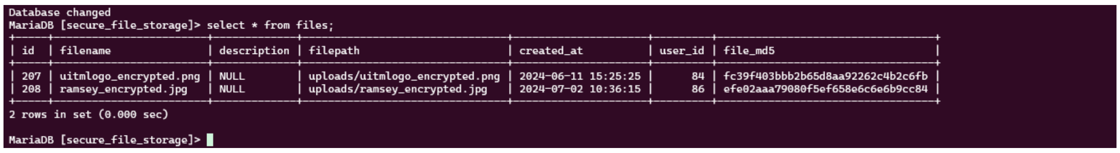
Files uploaded by the user are encrypted using the AES key before being stored on the server. This ensures that the files remain secure and cannot be accessed without proper decryption. The console output during encryption and decryption provides details about the process, including the MD5 hash of the original and decrypted files, encryption and decryption times, and the success status of file uploads.



Name: uitmlogo_encrypted.png
Size: 53,775 bytes
Type: unknown
Loaded: 100%

```
ed2d818fdc269125c28ed5ec6e066d70
```

Figure 4. 31 File Encrypted Using AES



```
Database changed
MariaDB [secure_file_storage]> select * from files;
+-----+-----------------------+-------------+-------------------------------+---------------------+---------+----------------------------------+
| id  | filename              | description | filepath                      | created_at          | user_id | file_md5                         |
+-----+-----------------------+-------------+-------------------------------+---------------------+---------+----------------------------------+
| 207 | uitmlogo_encrypted.png | NULL       | uploads/uitmlogo_encrypted.png | 2024-06-11 15:25:25 |      84 | fc39f403bbb2b65d8aa92262c4b2c6fb |
| 208 | ramsey_encrypted.jpg  | NULL        | uploads/ramsey_encrypted.jpg  | 2024-07-02 10:36:15 |      86 | efe02aaa79080f5ef658e6c6e6b9cc84 |
+-----+-----------------------+-------------+-------------------------------+---------------------+---------+----------------------------------+
2 rows in set (0.000 sec)

MariaDB [secure_file_storage]>
```

Figure 4. 32 File Encrypted Stored in Database MariaDB/MySQL

The storage of encrypted files on the server involves a multi-step process to ensure data security and integrity. The format and security measures of the encrypted file are highlighted in Figure 4.31, emphasizing the robustness of the encryption process. Finally, Figure 4.32 depicts how the encrypted files are stored in the MariaDB/MySQL database, illustrating the comprehensive security framework in place for data storage. These figures collectively provide a detailed overview of the encryption process, from client-side preparation to secure server storage, ensuring that user data remains protected throughout its lifecycle.

4.3.4 Data Integrity Verification

Ensuring data integrity is crucial for maintaining the reliability and security of a secure file storage system. This section outlines the process used to verify the integrity of data using MD5 hash functions. The primary objective is to verify that the data remains intact and unaltered during the encryption and decryption processes by comparing MD5 hash values.

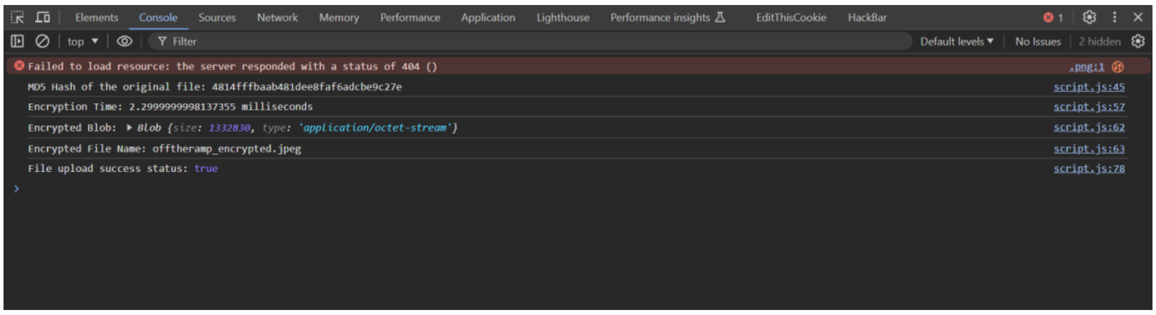
The method for verifying data integrity involves several key steps. First, MD5 hash values are generated for the original files before any encryption takes place. These hash values serve as a baseline for comparison. Next, the files are encrypted using the AES algorithm, and then decrypted back to their original form. After decryption, MD5 hash values are generated for the decrypted files. Finally, the MD5 hash values of the original and decrypted files are compared to ensure they match, indicating that the data has remained unchanged throughout the encryption and decryption processes.

Table 4. 3: MD5 Hash Comparisons for Various File Types and Sizes

File Type	File Size	MD5 Match (yes/no)
.txt	1KB	Yes
.pdf	100KB	Yes
.jpeg/png	1MB	Yes
.docx	10MB	yes

Table 4.3 indicate that the MD5 hash values for all tested file types and sizes match before and after the encryption and decryption processes. This consistency confirms that no data corruption occurred during these operations. The use of MD5 hash functions proved to be reliable for performing integrity checks, ensuring that the data remains secure and unaltered in the secure file storage system. This verification

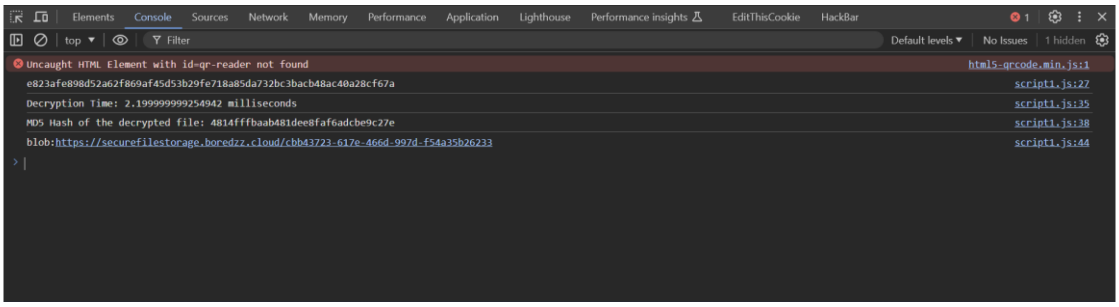
process is essential for maintaining trust in the system's ability to preserve the integrity of user data.



```
Failed to load resource: the server responded with a status of 404 ()            .png:1
MD5 Hash of the original file: 4814fffbaab481dee8faf6adcbe9c27e                  script.js:45
Encryption Time: 2.2999999998137355 milliseconds                                 script.js:57
Encrypted Blob: ▶ Blob {size: 1332830, type: 'application/octet-stream'}         script.js:62
Encrypted File Name: offtheramp_encrypted.jpeg                                   script.js:63
File upload success status: true                                                 script.js:78
>
```

Figure 4. 33 Encryption Console Output

Figure 4.33 depicts the console output during the encryption process. It shows the details of the encryption operation, including the time taken and the MD5 hash of the original file, ensuring that the process is correctly executed.



```
Uncaught HTML Element with id=qr-reader not found                                          html5-qrcode.min.js:1
e823afe898d52a62f869af45d53b29fe718a85da732bc3bacb48ac40a28cf67a                           script1.js:27
Decryption Time: 2.199999999254942 milliseconds                                            script1.js:35
MD5 Hash of the decrypted file: 4814fffbaab481dee8faf6adcbe9c27e                           script1.js:38
blob:https://securefilestorage.boredzz.cloud/cbb43723-617e-466d-997d-f54a35b26233          script1.js:44
> |
```

Figure 4. 34 Decryption Console Output

Figure 4.34 shows the console output during the decryption process. It includes the details of the decryption operation, such as the time taken and the MD5 hash comparison, verifying that the file has been accurately decrypted, and its integrity maintained.

4.3.5 Malware Scanning and Analysis

This is necessary to have a secure cloud storage where files being uploaded are not infected maliciously. This area tests the efficiency of the malware scan and analysis capability introduced into the system that makes use of VirusTotal API to identify recognized threats.

The assessment method contains several stages to certify the scanning action. After uploading a file, the user is subjected to a pre-upload component in order to scan the file first before the encrypted file could be stored. This element computes the hash of the file and uploads the hash to the VirusTotal API that scans the file against a bigger database of malware signatures. In other cases where the hash is not recognized or the file has not been analyzed before, the system will upload the file undergoes complete checking. Depending on the scan result, the system either permits or not the upload.

It was tested according to the following scenarios:

1. Uploading of an up-to-date file without threats being identified.
2. Uploaded a file that is definitely malicious (e.g. an EICAR test file).
3. Uploading a file that some earlier analysis was already done by VirusTotal.
4. On-the-fly scanning of an unknown file uploaded to the server.

Table 4. 4 Results of Malware Scanning and Analysis

Scenario	**Expected Result**	**Actual Result**
Uploading a clean file	File accepted	File accepted
Uploading a known malicious file (EICAR)	File rejected	File rejected
Uploading a previously analyzed safe file	File accepted instantly	File accepted instantly
Uploading a new unknown file	Scanned before upload	Scanned before upload

The system effectively blocks files detected as malicious by VirusTotal, ensuring that threats are not stored or processed. This pre-upload scanning acts as a protective layer to safeguard the server and other users from potential malware infections.

Additionally, logs are generated for every scanning activity, including the hash, file type, scan status, and result summary. This information is stored securely and can be used for auditing and security reviews.The evaluation confirms that the malware scanning and analysis feature successfully prevents malicious files from being stored in the system, aligning with industry-standard security practices and enhancing the trustworthiness of the platform.

4.4 Discussion

A malware scanning and analysis conducted in the secure file storage system will give first favourable defense against computer threats. Through the Virus Total API, the system can determine and block files with any known malware before they are even stored or encrypted. Such active method also makes sure that the evil such as trojan, ransomwares and spyware do not penetrate through the system, thus safeguarding both storage and end user.

The analysis of malware will be efficient and has less interference. One method of detecting previously known malicious files is by comparing the hash of the files with a large malware database via which the system can quickly find out whether the file has ever been recognized malicious before. In case of unknown or new files, they are scanned fully. The testing outcomes reveal that the framework can effectively prevent malicious files and enable clean files to move to the encryption process to offer appropriate and stable protection.

This malware filtering model accompanies the processing layer that offers secrecy and information security. After a malware scan has been passed, the file is encrypted with the AES-GCM algorithm, one of the modern and secure encryption algorithms that provide the data with confidentiality and integrity verification. The performance tests revealed that AES processes files of different size (between 1KB and 10MB) effectively, with insignificant encryption and decryption time. AES-GCM also has an authentication functionality to defend tampering.

Besides encryption the data integrity is also guaranteed by MD 5 hash encryption where the original file is compared with the decrypted file. This agrees that, the encryption-decryption procedure leaves the content unchanged, which further cements the reliability of the system.

Overall, using both malware scanning and AES encryption, and MD5 CC verification, the provided measures can be described as multitherapy approach. The system is able to provide protection against upload of malicious files as well as ensuring that encrypted and tamper-proof data have remained intact during the entire lifecycle.

4.5 Summary

The secure file storage system has not only integrated malware scanning of files before uploading through API VirusTotal but also added AES encryption to provide data security and confidentiality. Malware testing stops the malicious files to enter into system and AES encryption keep authorized files out of access of unknown users. The integrity of data is also checked through MD5 type hashing, verifying that there are no changes to any files during encryption or the decryption process. Such multilayered structure offers the system to be a secure and reliable system to work with sensitive data.