

# UNIVERSITI TEKNOLOGI MALAYSIA

## PRESENTATION SLIDE

Cloud Based Encryption Platform for Google  
Drive using Post-Quantum Cryptography  
Algorithm

YT Link: <https://youtu.be/o1BrbC1kaPM>



# Introduction & Problem Background



In the digital age, cloud computing technology, like Google Drive, offers unparalleled convenience for remote data storage and access. However, cybersecurity threats pose significant risks. Metomic's 2023 Google Scanner Report revealed that 40.2% of 6.5 million scanned Google Drive files contained sensitive data, making them susceptible to unauthorized access and data breaches. This vulnerability impacts users' trust and raises concerns about data integrity, confidentiality, and availability. The primary issue is the inadequacy of current encryption methods to protect against sophisticated cyber-attacks and the emerging threat of quantum computing. AES encryption, while effective now, may not withstand future quantum attacks. The National Institute of Standards and Technology (NIST) in the USA is developing post-quantum encryption standards, anticipating quantum supremacy within the next decade, which will render current asymmetrical algorithms like RSA ineffective (Clyde & Gillis, 2023). Addressing these deficiencies early is crucial to avoid higher costs and increased security risks in the future. Without robust encryption, sensitive data on platforms like Google Drive remains vulnerable to cybercriminals.

# PROJECT AIM

The aim of this project is to create a cloud encryption system for Google Drive which can be used to encrypt data stored in cloud and also to showcase the usage of post quantum algorithm in a Google Drive encryption system.

# OBJECTIVES

The objectives of the project are:

- (a) To study the current cloud based encryption platform that is used for files and existing post-quantum cryptography algorithms.
- (b) To design and develop a cloud-based encryption platform specifically for files with the addition of integrating post-quantum cryptography algorithms.
- (c) To validate the result by conducting security performance testing

# Project Scope

The scopes of the project are:

- (a) The stakeholder in this project will be Google Drive as the technology this project focus on is Google Drive
- (b) SafeDrive will cover only basic functionality like file encryption, decryption, and secure sharing of encrypted files in Google Drive
- (c) SafeDrive will be implementing post-quantum cryptography algorithm like NTRUEncrypt
- (d) SafeDrive will involve research into encryption techniques and cloud storage security systems. It will require significant coding effort to develop the encryption algorithms, user interface, and integration with Google Drive APIs. Testing will be important to ensure the security and functionality of the tool.
- (e) The targeted user for this project will be internet user who want to secure their files in Google Drive



1

Quantum computing, utilizing phenomena like superposition and entanglement, allows quantum bits (qubits) to exist in multiple states simultaneously, enabling parallel computation and significantly faster processing for certain tasks compared to classical computers. This poses a substantial threat to traditional encryption methods such as RSA and Diffie-Hellman, which rely on the difficulty of factorizing large numbers, as algorithms like Shor's can break these encryptions efficiently. Additionally, Grover's algorithm can halve the effective security of AES, making AES-256 as secure as AES-128. These vulnerabilities highlight the urgent need for post-quantum cryptography to safeguard data against potential quantum attacks (Liu, 2023; Patel, 2017; Bonnetain et al., 2019).

2

Post-Quantum Cryptography (PQC)

AlgorithmsPost-Quantum Cryptography (PQC)

focuses on ensuring cryptographic security against quantum computing threats, preparing current systems for a quantum era. Types of PQC include:

Lattice-Based Cryptography: Uses multidimensional lattices to solve cryptographic problems. Example: NTRUEncrypt.

Hash-Based Cryptography: Ensures message integrity using hash functions. Example: Merkle's hash-tree signature.

Code-Based Cryptography: Utilizes error-correcting codes to create secure encryption. Example: McEliece encryption.

Multivariate Public Key Cryptosystems (MPKC): Relies on sets of quadratic polynomials over finite fields for security. Example: Hidden Field Equation (HFE) scheme.

# Lattice-Based CryptoGraphy

Lattice-based cryptography stands out as a leading contender for securing data against quantum computing threats (Regev, 2006; Pradhan et al., 2019). A lattice is a grid of regularly spaced points in a vector space, represented by a basis of vectors.

These bases ensure unique lattice representations, crucial for cryptographic applications where security relies on solving complex lattice problems:



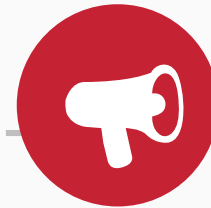
## Shortest Vector Problem (SVP)

Finds the shortest non-zero vector in a lattice, challenging even for quantum computers.



## Closest Vector Problem

Determines the lattice point closest to a given arbitrary point



## Learning With Error (LWE)

Hides secret information by introducing errors into equations, making decryption difficult (Lyubashevsky & Vadim, 2013).



## Shortest Independent Vector Problem

Seeks a set of short, linearly independent vectors within a lattice (Gama & Nguyen, 2008).

# Cloud Computing & Encryption

Cloud computing, widely adopted for its scalability and accessibility, relies on shared computing resources rather than local servers (Pankaj Madaan, Jagdep Singh). Security in cloud services like Google Drive and OneDrive is maintained through robust encryption methods such as AES and SSL/TLS, ensuring data confidentiality, integrity, and availability (Mulyadi, 2017). Despite these measures, concerns persist regarding data control and privacy, leading to the adoption of client-side encryption tools like Cryptomator and Boxcryptor. These tools add an additional layer of security by ensuring that decryption keys remain under user control rather than with the cloud provider.

# Cryptomator

**Cryptomator** is an open-source encryption software designed specifically for securing files stored in cloud storage services like Dropbox, Google Drive, and OneDrive.

**Encryption Method:** Utilizes AES encryption (256-bit) to encrypt files both at rest and during transmission.

**Client-Side Encryption:** Ensures that encryption and decryption occur locally on the user's device, maintaining full control over encryption keys.

**Platform Compatibility:** Supports multiple cloud storage providers and is available on various platforms including Windows, macOS, Linux, Android, and iOS.

**Open-Source:** Allows transparency and community contributions to the codebase, enhancing security through peer review.

**User Interface:** Offers a user-friendly interface for managing encrypted files and folders directly within cloud storage.

# BoxCryptor

**Boxcryptor**, developed by Secomb GmbH, is a commercial encryption software providing similar functionalities for securing cloud-stored files. Its key features include:

**Encryption Standard:** Utilizes AES encryption (256-bit) to encrypt files and folders stored in cloud services.

**Client-Side Encryption:** Encrypts data locally before uploading it to the cloud, ensuring that encryption keys remain under user control.

**Cross-Platform Support:** Available on multiple platforms such as Windows, macOS, Linux, Android, and iOS, supporting integration with various cloud storage providers.

**Security Audits:** Regularly undergoes security audits to ensure compliance and maintain high standards of encryption.

**Enterprise Solutions:** Offers solutions tailored for business and enterprise users, providing additional features like centralized management and compliance tools.



# Post-Quantum Cryptography (PQC)

Post-quantum cryptography (PQC) offers advanced security measures designed to withstand quantum computing threats. NTRUEncrypt, for example, employs polynomial-based encryption resistant to quantum attacks and demonstrates superior performance compared to traditional cryptographic algorithms like RSA and ECC (Cherckesova et al., 2020). As quantum computing capabilities advance, the adoption of PQC algorithms becomes increasingly critical to safeguard sensitive data against future threats.

# Security Measures in Google Drive

Google Drive, a leading cloud storage service, implements rigorous security measures to protect user data. These include strong encryption for data in transit and at rest, TLS for secure data transmission, and robust authentication mechanisms such as two-factor authentication (Abdulahi et al., 2019). Despite these efforts, ongoing research focuses on enhancing cloud security in response to evolving threats and privacy considerations within cloud environments (Takabi et al., 2010).

# NTRUEncrypt

Key parameters include  $N$ ,  $p$ , and  $q$ , where  $p$  and  $q$  must be mutually prime for algorithm strength.

## Key Generation:

Involves selecting small polynomials  $f$  and  $g$ , finding their inverses modulo  $p$  and  $q$ , and generating a secret key  $(f, fp)$  and public key  $h$ .

## Encryption:

Alice transforms a message into a polynomial  $m$  with coefficients modulo  $p$ , encrypts it using a small polynomial  $r$ , and calculates ciphertext  $e$ .

## Decryption:

Bob decrypts the received ciphertext  $e$  using his secret key  $f$ , recovering the original message.

NTRUEncrypt demonstrates superior speed and performance, being four times faster than RSA and three times faster than ECC (Cherckesova et al., 2020). Its resistance to quantum computing attacks and other cryptographic vulnerabilities makes it a promising choice for future-proof encryption. Would you like more details on how NTRUEncrypt compares to other cryptographic algorithms or its specific applications in secure communication?

1

2

3

4

NTRUEncrypt is a robust example of post-quantum cryptography, leveraging quadratic equations for encryption and decryption. Unlike traditional systems, it incorporates probabilistic encryption, offering multiple encryption options per message for enhanced security and efficiency. This method is known for its fast encryption and decryption processes, simplifying key production compared to RSA.

# System Dev Methodology

## 05. Deployment and Review

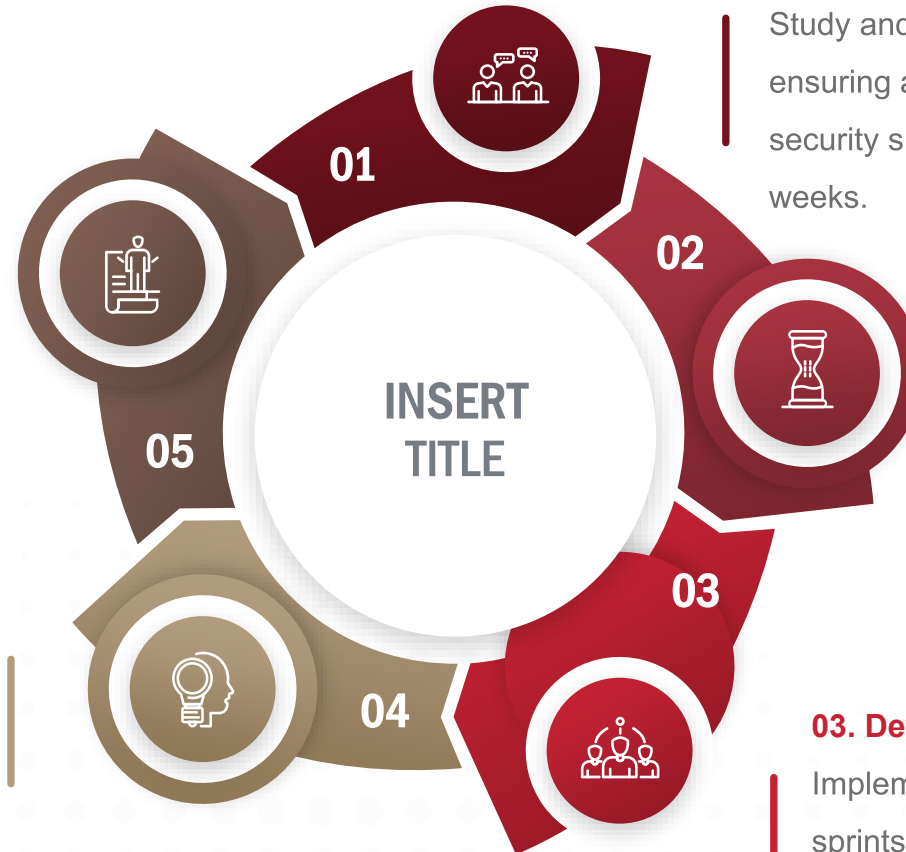
Prepare SafeDrive for public release, including user acceptance testing to verify reliability and functionality. Ongoing process post-deployment to address issues and ensure security compliance. Initial duration:

1 week

## 04. Testing Phase

Conduct comprehensive system and performance testing, including security tests like penetration testing. Duration:

Minimum 1 week..



## 01. Requirement Phase

Study and define project requirements, ensuring alignment with user needs and security specifications. Duration: 2 weeks.

## 02. Design Phase

Develop system architecture and user interface prototypes to ensure functionality and usability. Duration: 1 week..

## 03. Development Phase

Implement SafeDrive iteratively in sprints to meet functional requirements, focusing on Agile principles. Duration: Variable, depending on sprint phases.

# Technology Used Description

## • Front End Tech

- HTML, CSS, JavaScript: For creating interactive web interfaces.
- React: Framework for building UI components.

## BACK END TECH

- MySQL: Database management system.
- Python and Flask: Backend development for server-side logic.
- Pyntru: Python library for implementing NTRUEncrypt.
- Google Drive API: Integration for cloud storage functionality.

## • Hardware

- Processor: Intel Core i5 or equivalent.
- Memory (RAM): Minimum 8GB.
- Storage: 256GBSSD.
- Optional: Graphics card for enhanced processing (not essential)..

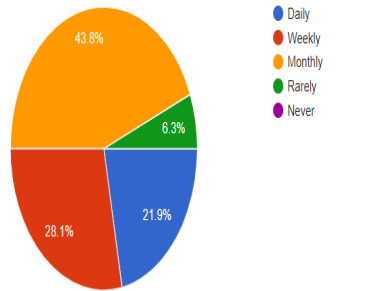
## • Software

- Operating System: Windows Server
- Web Server: Python 3.7+ with Flask.
- Database: MySQL for data storage.
- Encryption Libraries: Pyntru for NTRUEncrypt implementation.
- Cloud Services: Google Drive API for cloud storage integration.



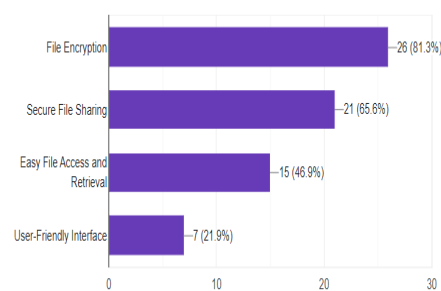
How often do you use Google Drive for storing and sharing files?

32 responses



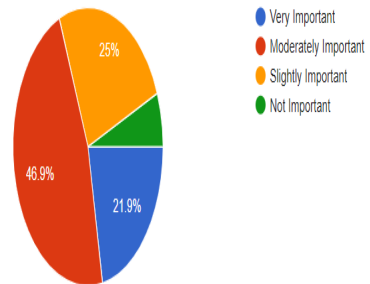
Which features are most important to you when using a cloud storage service like Google Drive? (Select all that apply)

32 responses



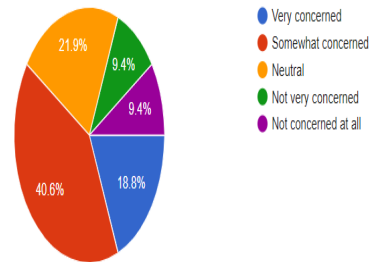
How important is it for you to share encrypted files with others securely

32 responses



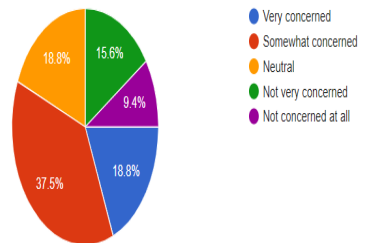
How concerned are you about the security of your files stored on Google Drive?

32 responses



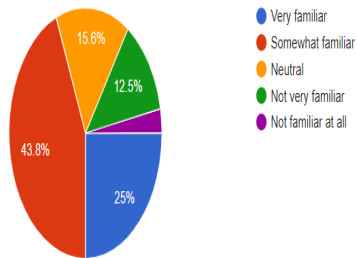
How concerned are you about the potential threat of quantum computing to current encryption methods (e.g., AES, RSA) used in cloud storage services like Google Drive?

32 responses



How familiar are you with the concept of quantum computing and its potential impact on data security?

32 responses



# Survey Result

Based on the survey, there are a few striking point that are worth mentioning, first is majority want file encryption and secure file sharing as primary features they focused on. Next is a sizeable majority of concerned of the security of their files in Google Drive. This shows when ask about its importance, a majority voted for importance of sharing files securely. In term of quantum computing and its threat, fortunately UTM students have the awareness of quantum computing and vote immensely on wanting a cloud encryption system with post quantum cryptography.

Based on the above, functional requirements are created to fulfill the expectation of user in the application.

Function	Requirement	Acceptance Criteria
User Login	As a Google Drive user, it is important to be able to access SafeDrive by integrating Goolge account with the system so that I do not have to create new, distinct account	<ul style="list-style-type: none"> <li>• User redirected to Safe Drive page</li> <li>• User can successfully log in using Google account</li> <li>• User redirected to SafeDrive after logging in</li> </ul>
File Encryption	As a google Drive user, the file need to be encrypted with NTRUEncrypt before being downloaded so that the files is secured.	<ul style="list-style-type: none"> <li>• User can select files from local storage</li> <li>• SafeDrive encrypt using NTRUEncrypt for encryption security</li> <li>• Encrypted files uploaded to Google Drive</li> <li>• SafeDrive notify successful completion and upload file</li> </ul>
Store and Manage Key	User want to securely store encryption key in password-protected database. This allow safe and secure access to encrypted file.	<ul style="list-style-type: none"> <li>• Encryption keys automatically stored in database protected by user password</li> <li>• User can retrieve and manage key through secure interface</li> <li>• Database use robust encryption to protect keys.</li> </ul>
Share Unlock Key	As a user, the unlock key can be shared through secure email so recipient can securely decrypt the shared file	<ul style="list-style-type: none"> <li>• User can generate decryption key for encrypted file</li> <li>• SafeDrive send email of the decryption key to recipient</li> <li>• Recipient can use the key to decrypt shared file using Crypt Drive</li> </ul>
Download and Decrypt Shared Files	As a recipient, it is important to be able to download the encrypted files from Google Drive and decrypt using decryption key sent via email.	<ul style="list-style-type: none"> <li>• Recipient receives email containing decryption key</li> <li>• Recipient downloads encrypted files from Google Drive</li> <li>• Recipient imports key into SafeDrive</li> <li>• SafeDrive decrypt files using imported key</li> <li>• Decrypted files saved into local storage</li> </ul>

# USE CASE DIAGRAM



Login Using Google Account: User logs into Safe Drive with their Google account.

Upload File: User encrypts a file and uploads

.Download File: User downloads an encrypted file and decrypts it with a key.

Encrypt File: User drags and encrypts a file in Safe

Decrypt File: User decrypts a file that was encrypted

Share Encryption Key via Email: User sends an encrypted key via email for file decryption

Generate Encryption Key: User generates a new encryption key

Store Encryption Key: User securely stores an encryption key

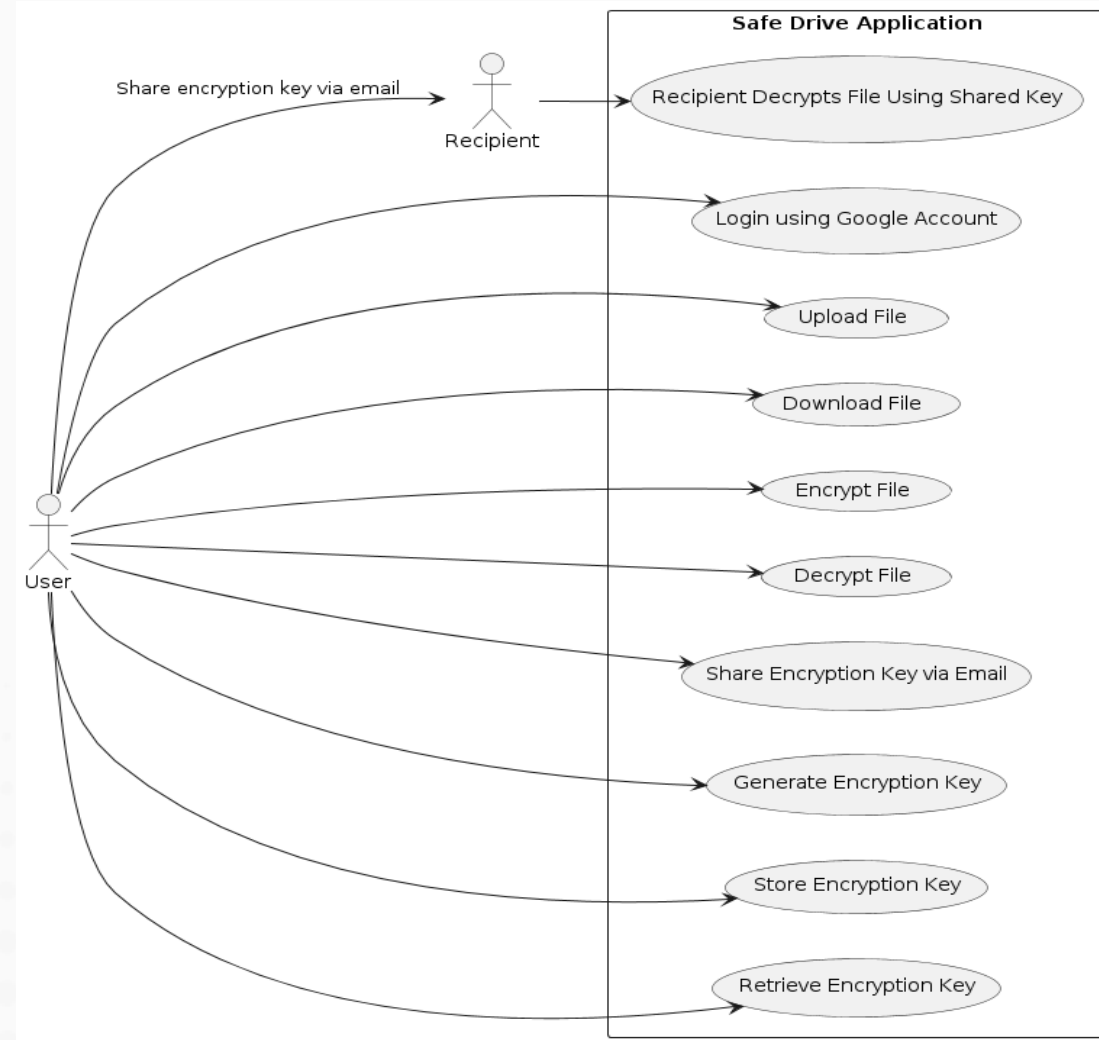
Retrieve Encryption Key: User retrieves an encryption key from Safe Drive

using an assigned ID.

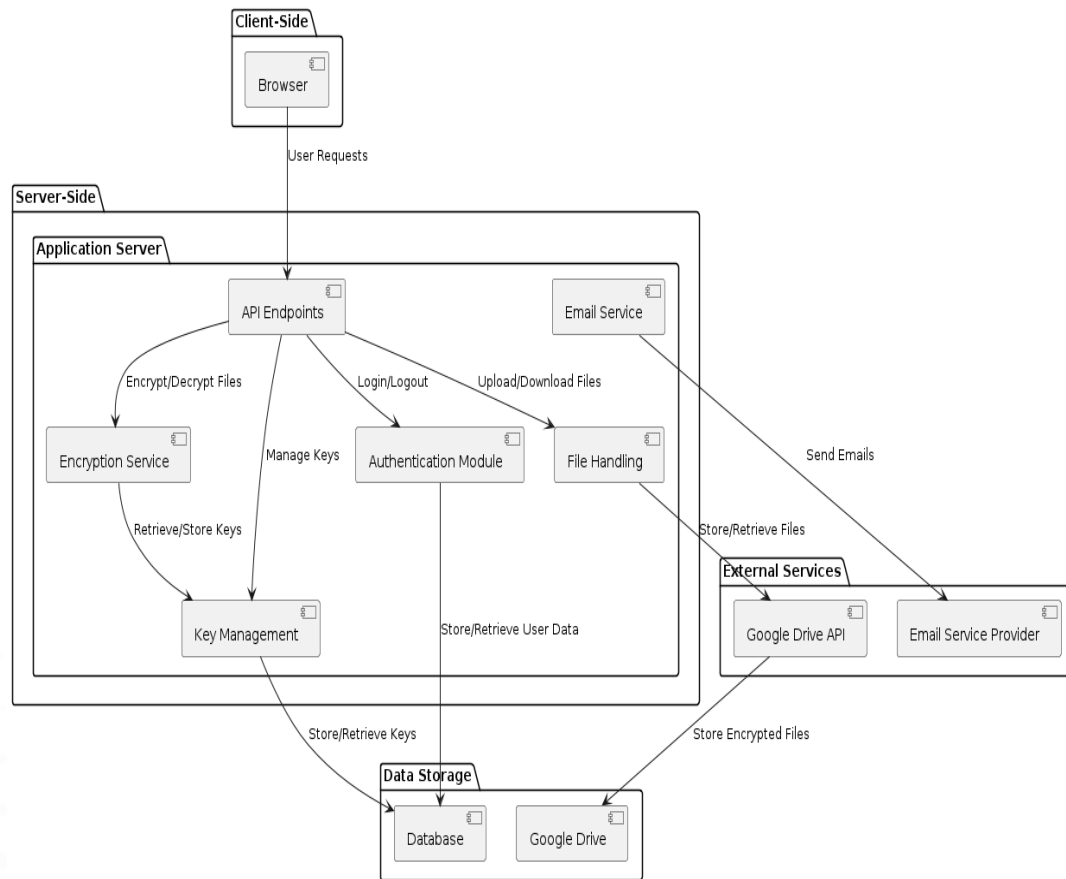
Recipient Decrypts File Using Shared Keys:

Recipient decrypts a file using a key shared via

email.



# System Architecture Design



System Architecture

**User Interface (UI):** Safe Drive is a more comprehensive tool and contains this part as well as the backend part that users do not see. These are the login screen, uploading and downloading files, the management of the encryption keys for files, and the shared files.

**Safe Drive Application:** This holds the actual business logic of the application. It handles the login of users, file encryption/decryption, stored files (which utilizes Google Drive API), generation of keys and storage, and email messaging for sharing of keys.

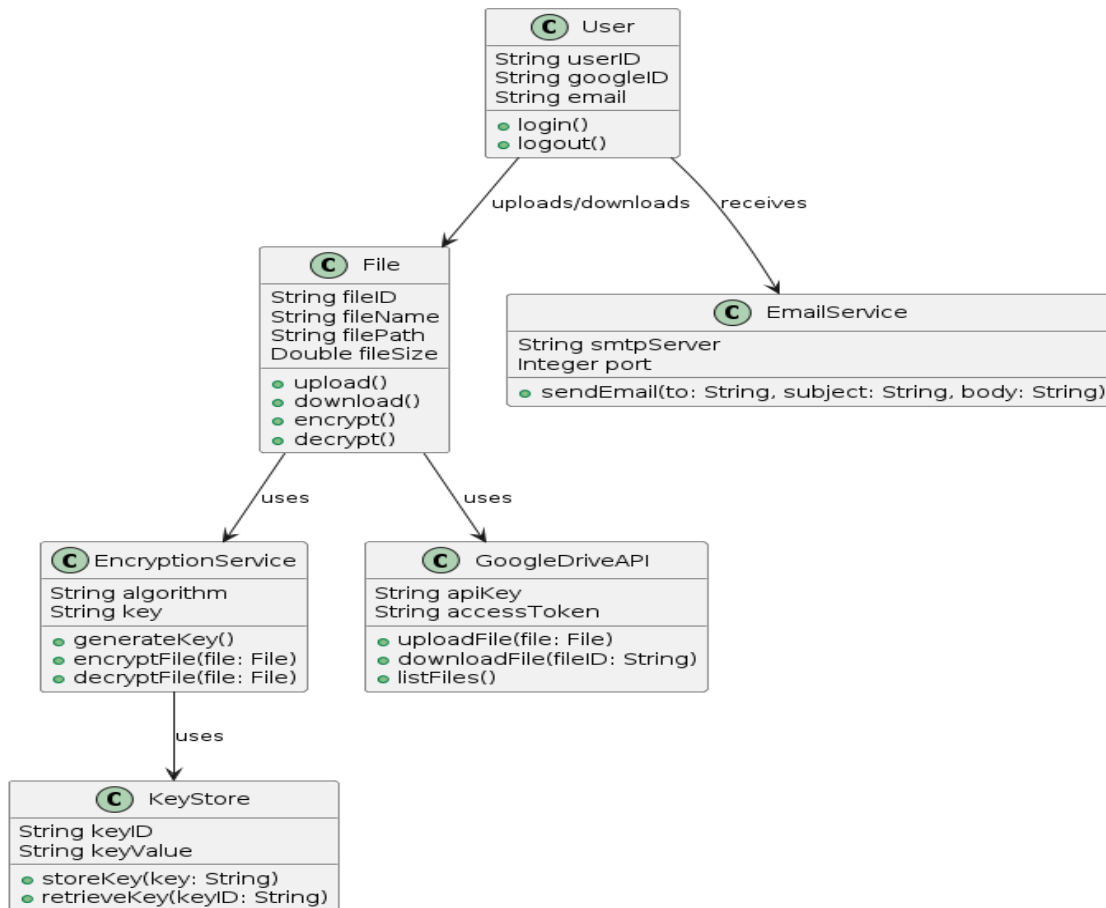
**Google Drive API Integration:** Safe Drive utilizes the Google Drive API to enable files to be saved in the cloud space linked with the application to ensure they are highly encrypted. This component is assigned for file uploading, downloading and list the files that are stored in Google Drive.

**Encryption Service:** It is required to encrypt and decrypt files using the selected encryption algorithms(NTRUEncrypt). They make sure files do not from getting lost or corrupt while in transit and when stored in the system.

**Key Management:** Is responsible for the generation, storage and retrieval of encryption keys.

**Email Service:** E-mails encryption key to other users so that communication can take place.

# Class Diagram



Class Diagram

## User

- Attributes: userID, googleID, email
- Save personal information of user that is linked to Google ID that is necessary only

## File

- Attributes: fileID, filename, filepath, filesize
- Encapsulate information about files in Google Drive

## EncryptionService

- Attributes: NTRUencrypt, key
- Control encryption/decryption part

## GoogleDriveAPI

- Attributes: apiKey, accessToken
- Manage authentication token, credential to interact with Google Drive API

## KeyManagement

- Attributes: keyID, keyValue
- Responsible for managing encryption keys

## Email Service

- Attributes: smtpServer, port
- Setting to use when sending email using secure connections



# DATA DESIGN

Due to the simple nature of Safe Drive, it is only necessary to have database for managing user information and encryption key only. Crypt Drive accesses a database for the user and key management where information like the user ID connected with Google ID, and the keys will be encrypted and stored. This helps in the provision of safe and effective mean of authenticating and managing keys.

On the other hand, items like file attributes, encryption algorithms and service configuration (Google Drive API and email server) are usually managed in memory or configuration files that does not require database storage. This help to focus only on the core functionalities of Safe Drive making it more straightforward and avoid unnecessary complexity making it more complicated to build

Username Database Schema

Column Name	Data Type	Constraint	Description
userID	INT	PRIMARY KEY	Unique identifier
Username	VARCHAR(50)	NOT NULL	Username for authentication
Password	VARCHAR(255)	NOT NULL	Password for authentication
Email	VARCHAR(100)	NOT NULL	Email address of user

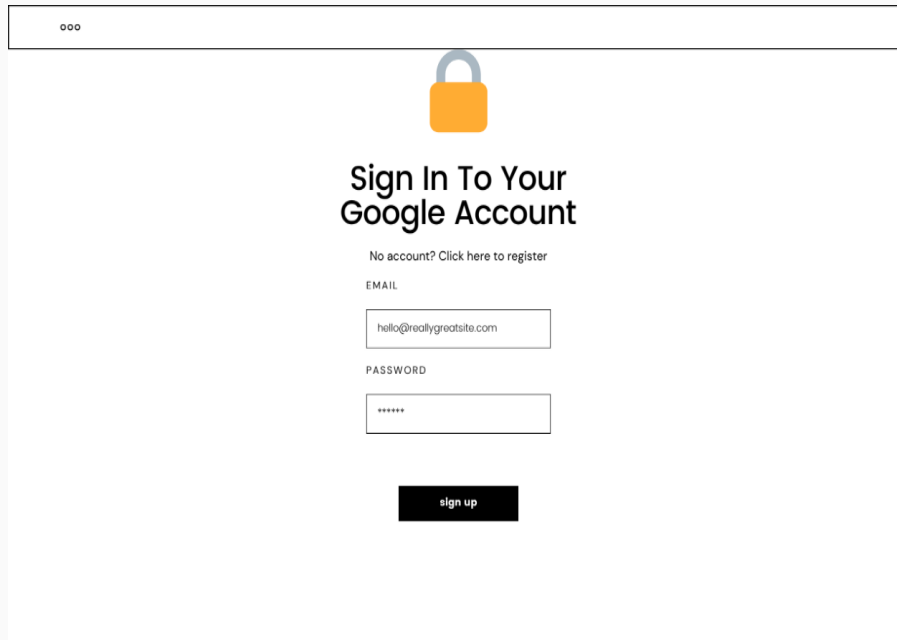
Key Management Database Schema

Column Name	Data Type	Constraint	Description
keyID	INT	PRIMARY KEY	Unique identifier
userID	INT	NOT NULL	Username for authentication
encryptionKey	VARCHAR(255)	NOT NULL	Encryption key for authentication


# Section Break

Suitable for all categories business and personal  
presentation farmers ensure that we will bring

# User Interface



ooo



**Sign In To Your Google Account**

No account? [Click here to register](#)

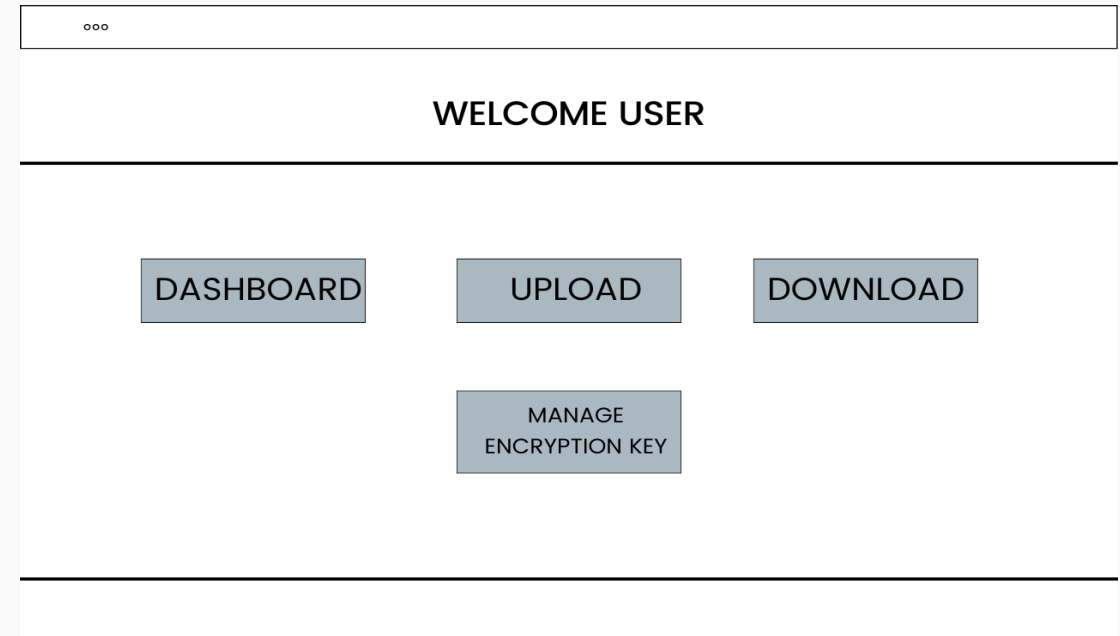
EMAIL

PASSWORD

[sign up](#)

01

Login Page



ooo

**WELCOME USER**

---

[DASHBOARD](#) [UPLOAD](#) [DOWNLOAD](#)

[MANAGE ENCRYPTION KEY](#)

---

02

SafeDrive Dashboard


# User Interface

ooo

UPLOAD FILES

SELECT FILE

UPLOAD FILE





SUBMIT



Progress

01 Upload Files

ooo

DOWNLOAD FILES

 FILE1.ENV 


 FILE2.ENV 

Progress

02 Download Files

# User Interface

...

Manage Encryption Key 

Generate New Key

Store Key

Stored Key

KEY 1


Retrieve

KEY 2


Retrieve

05 Manage Encryption Key

...

SEND EMAIL 

UPLOAD KEY



WRITE EMAIL

Email:TESTUSER1@GMAIL.COM

SEND

06 Send Email



# THANK YOU



In the Name of God for Mankind



[utm.my](http://utm.my)



[univteknologimalaysia](https://www.facebook.com/univteknologimalaysia)



[utmoofficial](https://www.instagram.com/utmoofficial)



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

***Menginovasi Penyelesaian***  
***Menginovasi Penyelesaian***  
***Menginovasi Penyelesaian***



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

***Innovating Solutions***  
***Innovating Solutions***  
***Innovating Solutions***