Module Name: Cyber Security Individual Project

Module Code: 7030CEM

Assignment Title: Project Brief (proposal)

Project Supervisor: Florence Nkosi

Project Supervisor Email: ae5962@coventry.ac.uk

Student Name: Khushal Divyang Patel

Student ID: 15405366

Assignment Due: 20/06/2025

Contents

Section A – ethics application

	I submitted my ethics application, and my application has been approved. I include my ethics certificate in the appendix as evidence.
o	I submitted my ethics application, and my application is currently under review.
	I have not submitted my ethics application.

Section B – project proposal

Project name: Enhancing data confidentiality and integrity in public cloud storage through client-side cryptographic overlay.

This project proposal outlines an initiative to enhance data confidentiality and integrity in public cloud storage through a client-side cryptographic overlay, addressing the critical issue of data privacy in commonly used cloud services.

1. Research question and problem statement

Research question: how can a client-side cryptographic overlay be effectively designed and implemented to provide true data confidentiality and integrity in public cloud storage, ensuring user control over data and keys while maintaining usability and robust key management?

Problem statement: the widespread adoption of public cloud storage services like google drive and Dropbox presents significant data privacy and security concerns. While these services offer convenience and scalability, users surrender direct control over their data's confidentiality. The core problem lies in the fact that cloud service providers technically possess access to user data, making it vulnerable to unauthorized access due to malicious employees, internal breaches, or external cyberattacks. Current security measures provided by these companies, though robust for their infrastructure, do not inherently prevent the provider or a compromised system from accessing unencrypted user data. This fundamental "trust issue" undermines user confidence and restricts the storage of highly sensitive information in public clouds. The lack of true end to-end user-controlled privacy is a critical gap that needs to be addressed.

Approach/method: the project will involve designing and developing a proof-of-concept client-side cryptographic overlay. This system will encrypt user files *before* they are transmitted to cloud storage, rendering them unreadable to the cloud provider or any unauthorized entity even if the cloud infrastructure is compromised. A central component of this approach will be the development of a novel, secure, and user-friendly key management system. This system will ensure that encryption keys are:

- **Super secure:** protected from unauthorized access by anyone, including the cloud provider.
- **Easy for the user to use** designed to prevent accidental lockouts and include robust recovery mechanisms.
- **Private from the cloud:** never exposed to the cloud service provider.

This research will explore the technical challenges of achieving this balance, particularly concerning secure key storage, distribution, and recovery in a decentralized and user centric manner.

2. Intended user or group of users and their requirements

Intended users: the primary intended users for this project are individuals and small to medium-sized enterprises (smes) who utilize public cloud storage services for personal or business data and are concerned about the privacy and security of their sensitive information. This includes, but is not limited to, professionals handling confidential client data, researchers storing proprietary findings, and general users seeking enhanced personal data privacy.

Need for this project: there is a pressing need for this project due to the inherent trust deficit in current public cloud storage models. Users are increasingly aware of data breaches and privacy concerns, yet the convenience of cloud storage remains appealing. This project directly addresses the desire for increased control and assurance over data confidentiality in the cloud without requiring users to host their own complex infrastructure. It provides a solution for those who wish to leverage the benefits of public cloud storage while mitigating the risks associated with provider access or compromise.
Needs of the intended user that the product should satisfy:

- **Confidentiality:** assurance that their files are unreadable by anyone other than themselves, including the cloud service provider.
- **Integrity:** confidence that their files have not been tampered with during storage or transmission.
- **Usability:** a system that is intuitive and integrates seamlessly with existing cloud storage workflows, minimizing user friction.
- **Key security:** a robust mechanism for storing and managing encryption keys that is both highly secure and accessible to the authorized user only.
- **Key recovery:** a reliable method to recover access to encrypted files even if the user's primary device or key storage is lost or compromised.

- **Cross-platform compatibility:** ideally, the solution should be adaptable to various operating systems and cloud providers.
- **Performance:** encryption and decryption processes should not significantly impede the user experience or file access times.

3. Systems requirements and final project outcome

Characteristics/properties of the final product: the final product, a client-side cryptographic overlay, should possess the following characteristics:

- **End-to-end encryption:** all data encrypted on the client device before upload and decrypted only on authorized client devices upon download.
- **Strong cryptography:** utilization of industry-standard, robust encryption algorithms (e.g., aes-256) and secure hashing functions.
- **Secure key management:** a system that generates, stores, and manages encryption keys securely, ensuring they are never transmitted to or stored by the cloud provider. This will include mechanisms for key derivation, key rotation, and secure backup/recovery.
- **Integrity verification:** implementation of mechanisms (e.g., hmac) to detect unauthorized modification of data stored in the cloud.
- **User authentication and authorization:** secure methods for authenticating the legitimate user and authorizing access to their encrypted files.
- **Overhead minimization:** designed to minimize performance impact on file upload/download speeds and storage consumption.
- **Modularity:** a modular architecture allowing for potential integration with different cloud storage APIs.

Final project outcome: the project aims to produce a robust proof-of-concept client-side cryptographic overlay that demonstrates the feasibility and benefits of enhancing data confidentiality and integrity in public cloud storage. This outcome will provide a practical framework and a functional prototype that clearly illustrates how users can maintain absolute control over their data's privacy in the cloud environment. The project will contribute to the understanding of secure and usable key management strategies for client-side encryption, addressing a critical gap in current cloud security paradigms. It will serve as a foundational step towards a more trust less cloud storage solution, where the cloud provider is merely a storage utility and not a data custodian with access to unencrypted content.

4. Primary research plan

This project's primary research will involve the design, implementation, and evaluation of a software demonstrator (the client-side cryptographic overlay with integrated key management). This method will allow for practical exploration of the challenges and effectiveness of the proposed solution.

Sequence of tasks/timeline:

Week 1: Literature Review & Threat Model Development

- Deliverable: Concise literature review on existing client-side encryption and key management methods.
- Threat model outlining key risks in public cloud storage.

Week 2: Requirements Specification & Architecture Design

- Deliverable: Clear system requirements (functional and non-functional).
- High-level architecture for the cryptographic overlay and key management.

Week 3–4: Cryptographic Module & Key Management Design

- Deliverable:
- Selection of encryption libraries and cryptographic schemes (e.g., AES256, HMAC).
- Design of a simplified but secure key management system with recovery features.

Week 5–6: Prototype Development

- Deliverable:
- A working prototype with basic encryption/decryption and file handling.
- Minimal UI/CLI for user interaction with files and keys.

Week 7: Integration, Security Testing, and Performance Evaluation

- Deliverable:
- Integrated system test (functional + security).
- Initial performance benchmarks (e.g., encryption time, CPU usage).

Week 8: Refinement, Documentation & Final Report

- Deliverable:
- Refined prototype based on feedback.
- Final project report, user guide, and technical documentation.

Data collection: data for evaluation will primarily be quantitative, focusing on performance metrics (e.g., encryption/decryption time in milliseconds, CPU usage in percentage, memory footprint in mb) collected during the testing phases. Qualitative data might be gathered through a small-scale usability assessment if conducted. The "amount of data" needed will correspond to varying file sizes to thoroughly test the system's efficiency.

Assumptions and scope: it is assumed that the project will focus on a single cloud storage provider for the proof-of-concept due to time constraints, although the design will aim for modularity. The key management system will prioritize security and recovery mechanisms that are practical for individual users. "over-promising" results will be avoided; the focus will be on demonstrating a functional and secure foundation for client-side encryption and key management, acknowledging that a full, production-ready system is beyond the scope of a single MSc dissertation.

5. Initial/mini literature review

This mini-literature review identifies key research areas and gaps that the project aims to address.

Client-side encryption in cloud storage:

- Research into existing client-side encryption solutions often highlights the trade-off between security and usability. For instance, solutions like cryptomator and boxcryptor provide client-side encryption, but their underlying key management schemes or reliance on user-managed master passwords can still present usability challenges or single points of failure. My project will build upon the principles of these tools while specifically innovating in the key management aspect to enhance both security and user convenience.

Secure key management for end-users:

- The literature on key management for end-users reveals significant challenges. Traditional public key infrastructure (pki) can be complex for average users, while simpler password-based systems are vulnerable to brute-force attacks. Research by authors such as [identify a relevant author, e.g., Schneier, or a specific paper focusing on usability in crypto] demonstrates the difficulty of designing systems where keys are truly user controlled, secure, and recoverable without relying on a central authority. My research will critically evaluate different approaches to decentralized or highly secure personal key management to find a balance between these competing requirements.

Data integrity in cloud storage:

- Beyond confidentiality, ensuring data integrity in cloud storage is crucial. While cloud providers implement internal integrity checks, the user lacks independent verification that their data has not been subtly altered by a malicious actor or system error. Research on verifiable storage and provable data possession (pdp) addresses this but often requires significant computational overhead or complex protocols. My project will explore lightweight mechanisms to integrate integrity checks alongside the encryption, potentially drawing from the principles of message authentication codes (macs) or digital signatures.

6. Gaps and contributions

Existing literature demonstrates a strong understanding of cryptographic primitives. However, a significant gap remains in the practical implementation of truly user-centric and robust key management systems for client-side cloud encryption that are simultaneously highly secure, resilient to single points of failure, and user-friendly for non-technical individuals. My project aims to fill this gap by proposing and prototyping a novel approach to key management that provides users with greater control and recovery options without compromising the security of their encryption keys or relying on the cloud provider for key custody. This critical and evaluative perspective on the existing literature highlights the necessity and importance of my research project.

7. Tools and methodology

System development life cycle (SDLC)	Agile
Purpose	Tools/Libraries/Methods
Cryptography	cryptography, PyCryptodome, hashlib, Fernet, HMAC, PBKDF2, AES-256
Dev Environment	Python, VS Code, Git, GitHub
Cloud	boto3, google-Api-python-client, Dropbox
UI/CLI	argparse, click, Tkinter, customtkinter

Key Management	keyring, sqlite3, secure password hashing, etc

Bibliography

1. Dholakia, S. (2023) *Modern Cryptography: The Practical Guide*. Independently published.
2. Katz, J. and Lindell, Y. (2021) *Introduction to Modern Cryptography*. 3rd edn. CRC Press.
3. Schneier, B. (2015) *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. 20th Anniversary edn. Wiley.
4. Aumasson, J.P. (2020) *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press.
5. Ristenpart, T., Tromer, E., Shacham, H. and Savage, S. (2009) ‘Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds’, *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, pp. 199–212.
6. Google Cloud (2024) *Client-side encryption*. Available at: https://cloud.google.com/docs/security/encryption/client-side
7. Google Cloud (2024) *Cloud Key Management Service documentation*. Available at: https://cloud.google.com/kms/docs
8. Cosmian (2024) *Secure your data in SaaS with Client-Side Encryption*. Available at: https://www.cosmian.com/resources/blog/secure-data-saas-clientsideencryption
9. Microsoft Q&A (2024) *Client-side encryption vs Client-side key encryption*. Available at: https://learn.microsoft.com/en-us/answers/questions/1704880
10. Reddit (2023) *Best books for learning cryptography?* Available at: https://www.reddit.com/r/cryptography/comments/j94kcy/best_books_for_learning_cryptography