**Coventry University**

**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**

1. **Section A - ethics application**
2. **Section B - project proposal**
   1. Research question and problem statement
   2. Intended user or group of users and their requirements
   3. Systems requirements and final project outcome
   4. Primary research plan
   5. Initial/mini literature review
   6. Gaps and contributions
   7. Tools and methodology
3. **Bibliography**

**Section A - ethics application**

|  |  |
| --- | --- |
|  |  |
| X | 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 title:** Enhancing data confidentiality and integrity in public cloud storage through a 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 as a web application. This system will feature a modern web interface where all cryptographic operations occur on the client-side. User files will be encrypted in the browser using the **Web Crypto API** before they are transmitted to a backend server, ensuring a zero-knowledge architecture.

For confidentiality and integrity, the system will utilize **AES-GCM (Galois/Counter Mode)**, an authenticated encryption mode that provides both encryption and data authentication simultaneously. Keys will be derived from user-provided recovery phrases using **PBKDF2**, a standard key derivation function that protects against brute-force attacks. The backend, developed using the **Flask framework**, will serve as a RESTful API for managing user accounts and storing the encrypted file metadata and content (as BLOBs) in a **SQLite** database, but it will never have access to plaintext data or encryption keys. This approach directly addresses the critical balance between maintaining usability and providing robust, user-controlled security.

**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:** Based on a review of existing literature concerning cloud security usability and an analysis of features in current privacy-enhancing tools, the product should satisfy the following core user requirements:

* **Confidentiality:** assurance that their files are unreadable by anyone other than themselves, including the 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 into a web-based workflow, 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.
* **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 web application, should possess the following characteristics:

* **Zero-Knowledge Architecture:** All data is encrypted on the client device (in-browser) before upload and decrypted only on authorized client devices upon download.
* **Strong Cryptography:** Utilization of industry-standard, robust cryptographic implementations, specifically **AES-GCM** for authenticated encryption and **PBKDF2** for key derivation.
* **Secure Key Management:** A system that generates, derives, and manages encryption keys securely on the client-side, ensuring they are never transmitted to or stored by the server. This will include mechanisms for key wrapping to securely store file keys.
* **Integrity Verification:** Inherent implementation of data integrity checks through the use of the AES-GCM authenticated encryption mode.
* **Web-Based Interface:** A modern, responsive user interface built with **Tailwind CSS** that provides a seamless user experience for file operations.
* **RESTful API Backend:** A **Flask**-based backend to handle user authentication and the storage of encrypted file metadata and content.

**Final project outcome:** the project aims to produce a robust proof-of-concept client-side cryptographic web application that demonstrates the feasibility and benefits of a zero-knowledge approach to 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. 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.

**4. Primary research plan**

This project's primary research will involve the design, implementation, and evaluation of a software demonstrator (the client-side cryptographic web application). 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 client-side encryption and key management. Threat model for web-based cryptographic applications.
* **Week 2:** Requirements Specification & Architecture Design
  + **Deliverable:** Clear system requirements. High-level architecture for the web application, detailing frontend-backend interaction and API endpoints.
* **Week 3-4:** Backend and Database Development
  + **Deliverable:** A functional Flask API with RESTful endpoints for user management and file metadata storage using SQLite.
* **Week 5-6:** Frontend and Cryptography Development
  + **Deliverable:** A working web interface using Tailwind CSS. Implementation of client-side encryption/decryption using the Web Crypto API (AES-GCM, PBKDF2).
* **Week 7:** Integration, Testing, and Evaluation
  + **Deliverable:** Fully integrated system test (functional, security, and usability). Usability testing with mock users to evaluate the key management and recovery flows.
* **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 be both quantitative and qualitative. **Quantitative data** will focus on performance metrics (e.g., encryption/decryption time, CPU usage). **Qualitative data** will be gathered through a small-scale usability assessment with mock users.

**Assumptions and scope:** The project will focus on building a self-contained proof-of-concept where the Flask backend also handles storage. The key management system will prioritize security and recovery mechanisms that are practical for individual users within a web-based context.

**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 within a web application context.

**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 foundational experts like **Schneier (2015)** demonstrates the inherent difficulty of designing systems where keys are truly user-controlled, secure, and recoverable without relying on a central, trusted authority. This highlights the "usability-security trade-off," which is a central challenge in this domain and a key focus of my research project.

**Data integrity in cloud storage:** Beyond confidentiality, ensuring data integrity is crucial. While cloud providers implement internal integrity checks, the user lacks independent verification. My project addresses this directly by using **AES-GCM**, an Authenticated Encryption with Associated Data (AEAD) cipher, which combines confidentiality and integrity into a single, efficient cryptographic operation, making separate integrity checks like HMAC unnecessary for the encrypted payload.

**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 within a web application, providing users with greater control and recovery options without compromising the security of their encryption keys.

**7. Tools and methodology**

|  |  |
| --- | --- |
| **Category** | **Tools/Technologies** |
| **System development life cycle (SDLC)** | Agile |
| **Frontend (Client-Side)** | JavaScript, Web Crypto API, Tailwind CSS |
| **Backend (Server-Side)** | Python, Flask, Werkzeug |
| **Database** | SQLite |
| **Cryptography** | AES-GCM, PBKDF2 |
| **Dev Environment** | VS Code, Git, GitHub |

**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](https://www.google.com/search?q=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/](https://www.google.com/search?q=https://www.reddit.com/r/cryptography/comments/j94kcy/best_books_for_learning_cryptography/)