Department of Software Engineering
BAHRIA UNIVERSITY KARACHI CAMPUS

Discovering Knowledge

COURSE: SEL 401
CLOUD COMPUTING
PROJECT REPORT

CLASS: BSE – 6 B (SPRING - 2025)

Data Vault

Group Members

Student Name	Enrollment#
Syed Ahnaf Raza (Group Leader)	02-131222-085
Seemab Asghar	02-131222-047
Memoona Iqbal	02-131222-062
Izaan Abdullah	02-131222-077

Submitted to:

Course Instructor: **Engr. Mahawish Fatima**
Lab Instructor: **Engr. Noor us Sabah**

Abstract

DataVault is a comprehensive, open-source, web-based encryption platform engineered to provide **military-grade data security** while maintaining **exceptional user accessibility**. It is purpose-built to address the growing global demand for **reliable, user-friendly data protection** in the face of increasing cyber threats, data breaches, and strict regulatory frameworks such as **GDPR**, and other regional compliance standards.

At its core, DataVault employs a **zero-knowledge architecture**, ensuring that all encryption and decryption occur entirely on the client side, and that no sensitive keys or plaintext data are ever transmitted to or stored on the server. The platform integrates **AES-256 symmetric encryption** and **RSA-2048/4096 asymmetric cryptography**, offering users the flexibility to secure **text, and documents, multimedia files,** with uncompromised confidentiality and integrity. Support for multiple cipher block modes—including **Cipher Feedback (CFB)**, **Cipher Block Chaining (CBC)**, and **Galois/Counter Mode (GCM)**—further strengthens security while accommodating diverse use-case requirements.

DataVault's robust backend is powered by **FastAPI**, ensuring asynchronous, high-throughput, and scalable API performance. The frontend, built with **Next.js**, delivers a responsive and seamless user experience across devices. **Docker-based containerization** enables rapid deployment, horizontal scalability, and easy integration with existing enterprise infrastructure. Key user-centric features include **drag-and-drop file encryption**, **real-time text encryption**, **key-pair generation**, and **stream-based file handling**, minimizing processing overhead and improving efficiency, especially in bandwidth-constrained environments.

Designed with a **modular, extensible architecture**, DataVault is future-proof and adaptable to emerging security paradigms. Planned enhancements include **cryptographic algorithms** to safeguard against quantum computing threats. The platform is already suitable for deployment in critical sectors such as **healthcare**, **legal services**, **government**, and **finance**, where confidentiality, auditability, and compliance are paramount.

By combining **transparency, open-source auditability, platform independence**, and **an intuitive design philosophy**, DataVault redefines digital security standards. It empowers individuals, organizations, and institutions to take control of their data privacy and integrity, ultimately contributing to a more secure and trustworthy global digital ecosystem.

TABLE OF CONTENTS

1. Introduction

DataVault is a comprehensive, web-based encryption platform designed to provide military-grade security for both individual users and organizations. In an era where data breaches and cyber threats are increasingly sophisticated, DataVault offers a robust solution for protecting sensitive information through advanced cryptographic techniques.

The platform implements multiple encryption standards, including AES-256 for symmetric encryption and RSA for asymmetric encryption, providing users with flexible options for securing their data. Built with modern web technologies, DataVault ensures accessibility across all devices while maintaining the highest security standards.

1.1.1 Key Objectives

- **Security First**: Implement industry-standard encryption algorithms with zero-knowledge architecture.
- **User-Friendly**: Provide an intuitive interface that makes advanced cryptography accessible to all users.
- **Versatility**: Support multiple encryption modes and file types for diverse use cases.
- **Transparency**: Open-source approach with clear documentation and audit trails.
- **Performance**: Optimize for speed without compromising security.
- **Scalability**: Design for both individual and enterprise-level usage.

1.1.2 Target Audience

- **Individual Users**: Privacy-conscious individuals protecting personal documents.
- **Business Professionals**: Securing confidential business communications and files.
- **Healthcare Organizations**: Protecting patient data and medical records.
- **Legal Firms**: Securing sensitive legal documents and client information.
- **Educational Institutions**: Protecting research data and student information.
- **Government Agencies**: Securing classified and sensitive government data.

2. Problem Statement

In today's digital landscape, data security is a critical concern for individuals and organizations. Several key challenges highlight the urgent need for a comprehensive encryption solution.

2.1 Current Security Challenges

Data Breach Epidemic

- Over 4.1 billion records exposed in data breaches in 2019 alone.
- Average cost of a data breach reached $4.24 million in 2021.
- 95% of successful cyberattacks are due to human error.
- Traditional security measures are insufficient against modern threats.

Complexity of Existing Solutions

- Most encryption tools require technical expertise.
- Command-line interfaces intimidate non-technical users.
- Lack of standardization across different platforms.
- Poor user experience leads to security shortcuts.

Accessibility Issues

- Enterprise-grade encryption tools are expensive.
- Limited cross-platform compatibility.
- Lack of real-time collaboration features.
- No unified solution for different encryption needs.

Trust and Transparency Concerns

- Closed-source solutions lack transparency.
- Vendor lock-in with proprietary formats.
- Unclear data handling practices.
- Limited audit capabilities.

2.2 Specific Problems Addressed

File Security

- Unencrypted files stored on cloud services.
- Lack of end-to-end encryption for file sharing.
- No integrity verification for transmitted files.
- Difficulty in managing encryption keys.

Communication Security

- Plain text communications vulnerable to interception.
- Lack of digital signatures for authenticity.
- No secure method for sharing encryption keys.
- Absence of forward secrecy in communications.

Key Management

- Poor key generation practices.
- Insecure key storage and distribution.
- Lack of key rotation policies.
- No centralized key management for organizations.

3. Proposed Solution

DataVault addresses these challenges through a comprehensive, web-based encryption platform that combines security, usability, and accessibility. The solution provides multiple encryption options while maintaining a user-friendly interface that requires no technical expertise.

3.1. Features of the project

3.1.1 Core Encryption Features

Advanced Encryption Standard (AES) Implementation

- **AES-256 Encryption**: Military-grade symmetric encryption with 256-bit keys.
- **Multiple Cipher Modes**:
  - *CFB (Cipher Feedback)*: Stream cipher mode for real-time data encryption.
  - *CBC (Cipher Block Chaining)*: Block cipher mode with high security.
  - *GCM (Galois/Counter Mode)*: Authenticated encryption with built-in integrity.
- **PBKDF2 Key Derivation**: Secure key derivation with configurable iterations (10,000 to 1,000,000).
- **Salt Generation**: Cryptographically secure random salt for each encryption operation.
- **HMAC Integrity Verification**: Optional SHA-256 HMAC for data integrity assurance.

RSA Public Key Cryptography

- **Hybrid Encryption**: Combines RSA and AES for optimal security and performance.
- **Key Pair Generation**: Support for 2048, 3072, and 4096-bit RSA keys.
- **Digital Signatures**: RSA-PSS signatures for authenticity verification.
- **Secure Key Exchange**: Public key infrastructure for secure communication.

Text Encryption Capabilities

- **Real-time Encryption**: Instant encryption of text messages and documents.
- **Unicode Support**: Full support for international characters and emojis.
- **Compression Options**: Configurable compression levels (None, Fast, Default, Best).
- **Batch Processing**: Ability to encrypt multiple text blocks simultaneously.

File Encryption System

- **Universal File Support**: Encrypt any file type (documents, images, videos, archives).
- **Large File Handling**: Efficient processing of files up to 100MB.
- **Metadata Preservation**: Maintain original file information and timestamps.
- **Binary Data Support**: Secure handling of binary and executable files.

3.1.2 Security Features

Zero-Knowledge Architecture

- **Client-Side Processing**: All encryption operations performed in the browser.
- **No Server-Side Key Storage**: Keys never transmitted to or stored on servers.
- **Temporary File Handling**: Automatic cleanup of temporary files.
- **Memory Protection**: Secure memory handling to prevent key leakage.

Advanced Security Measures

- **Secure Random Generation**: Cryptographically secure random number generation.
- **Key Stretching**: PBKDF2 with high iteration counts to resist brute force.
- **Forward Secrecy**: Each encryption operation uses unique salts and IVs.
- **Side-Channel Resistance**: Implementation resistant to timing attacks.

Integrity and Authentication

- **HMAC Verification**: Optional message authentication codes.
- **Digital Signatures**: RSA-PSS signatures for non-repudiation.
- **Checksum Validation**: SHA-256 checksums for file integrity.
- **Tamper Detection**: Automatic detection of data modification.

3.1.3 User Experience Features

Intuitive Interface Design

- **Modern UI/UX**: Clean, responsive design with dark/light theme support.
- **Drag-and-Drop**: Easy file upload with visual feedback.
- **Progress Indicators**: Real-time progress bars for long operations.
- **Error Handling**: Clear error messages with suggested solutions.

Accessibility Features

- **Keyboard Navigation**: Complete keyboard accessibility.
- **Mobile Responsive**: Optimized for smartphones and tablets.

Advanced Tools

- **Key Generator**: Secure random key generation with customizable length.
- **Encryption History**: Track recent encryption operations.
- **Settings Management**: Persistent user preferences and configurations.
- **Export/Import**: Backup and restore encryption settings.

3.1.4 Performance Features

Optimization Techniques

- **Streaming Processing**: Handle large files without memory overflow.
- **Chunked Upload**: Break large files into manageable chunks.
- **Background Processing**: Non-blocking operations with web workers.
- **Caching Strategy**: Intelligent caching for improved performance.

3.2. Methodology

Agile Development Approach The project follows an Agile methodology with iterative development cycles, continuous integration, and regular stakeholder feedback to ensure rapid delivery of functional features while maintaining high quality.

Security-First Development

- **Threat Modeling**: Comprehensive analysis of potential security threats.
- **Secure Coding Practices**: Implementation following OWASP guidelines.
- **Regular Security Audits**: Continuous security assessment and penetration testing.
- **Code Reviews**: Mandatory peer review for security-critical code.

Test-Driven Development (TDD)

- **Unit Testing**: Comprehensive test coverage for all functions.
- **Integration Testing**: End-to-end testing of encryption workflows.
- **Security Testing**: Specialized tests for cryptographic implementations.
- **Performance Testing**: Load testing and benchmarking.

3.2.2 Cryptographic Methodology

Industry Standard Algorithms

- **NIST Approved**: All algorithms approved by NIST.
- **Peer Reviewed**: Use of well-established, academically reviewed cryptographic methods.
- **Implementation Standards**: Following RFC specifications and best practices.
- **Regular Updates**: Keeping up with the latest cryptographic research.

Key Management Strategy

- **Secure Generation**: Use of cryptographically secure random number generators.
- **Proper Storage**: Client-side key storage with secure memory handling.
- **Key Rotation**: Support for regular key updates and rotation.
- **Backup and Recovery**: Secure key backup and recovery mechanisms.

3.2.3 Quality Assurance Methodology

Code Quality Standards

- **Static Analysis**: Automated code quality and security analysis.
- **Code Coverage**: Minimum 90% test coverage requirement.
- **Documentation**: Comprehensive inline and external documentation.
- **Version Control**: Git-based version control with detailed commit messages.

3.3. Technologies

3.3.1 Frontend Technologies

Next.js 14 Framework

- **Server-Side Rendering (SSR)**: Improved performance and SEO.
- **Static Site Generation (SSG)**: Pre-built pages for optimal loading.
- **API Routes**: Built-in API functionality for seamless integration.
- **Image Optimization**: Automatic image optimization and lazy loading.
- **TypeScript Support**: Built-in TypeScript support for type safety.

React 18 Library

- **Component-Based Architecture**: Modular, reusable UI components.
- **Hooks System**: Modern state management with React hooks.
- **Concurrent Features**: Improved performance with concurrent rendering.
- **Suspense**: Better loading states and error boundaries.
- **Context API**: Global state management for user preferences.

TypeScript Implementation

- **Type Safety**: Compile-time error detection and prevention.
- **IntelliSense**: Enhanced development experience with auto-completion.
- **Refactoring Support**: Safe code refactoring with type checking.
- **Documentation**: Self-documenting code with type annotations.
- **Team Collaboration**: Improved code consistency across team members.

Tailwind CSS Styling

- **Utility-First**: Rapid UI development with utility classes.
- **Responsive Design**: Mobile-first responsive design system.
- **Dark Mode**: Built-in dark mode support with theme switching.
- **Custom Components**: Reusable component library with shadcn/ui.
- **Performance**: Optimized CSS with unused style purging.

3.3.2 Backend Technologies

FastAPI Framework

- **High Performance**: One of the fastest Python frameworks available.
- **Automatic Documentation**: Auto-generated OpenAPI/Swagger documentation.
- **Type Hints**: Native Python type hint support for better code quality.
- **Async Support**: Full asynchronous request handling.
- **Validation**: Automatic request/response validation with Pydantic.

Python 3.11+ Runtime

- **Performance**: Latest Python version with improved performance.
- **Type System**: Enhanced type system for better code reliability.
- **Security**: Latest security patches and improvements.
- **Library Ecosystem**: Access to extensive Python library ecosystem.
- **Cross-Platform**: Consistent behavior across different operating systems.

Cryptography Library

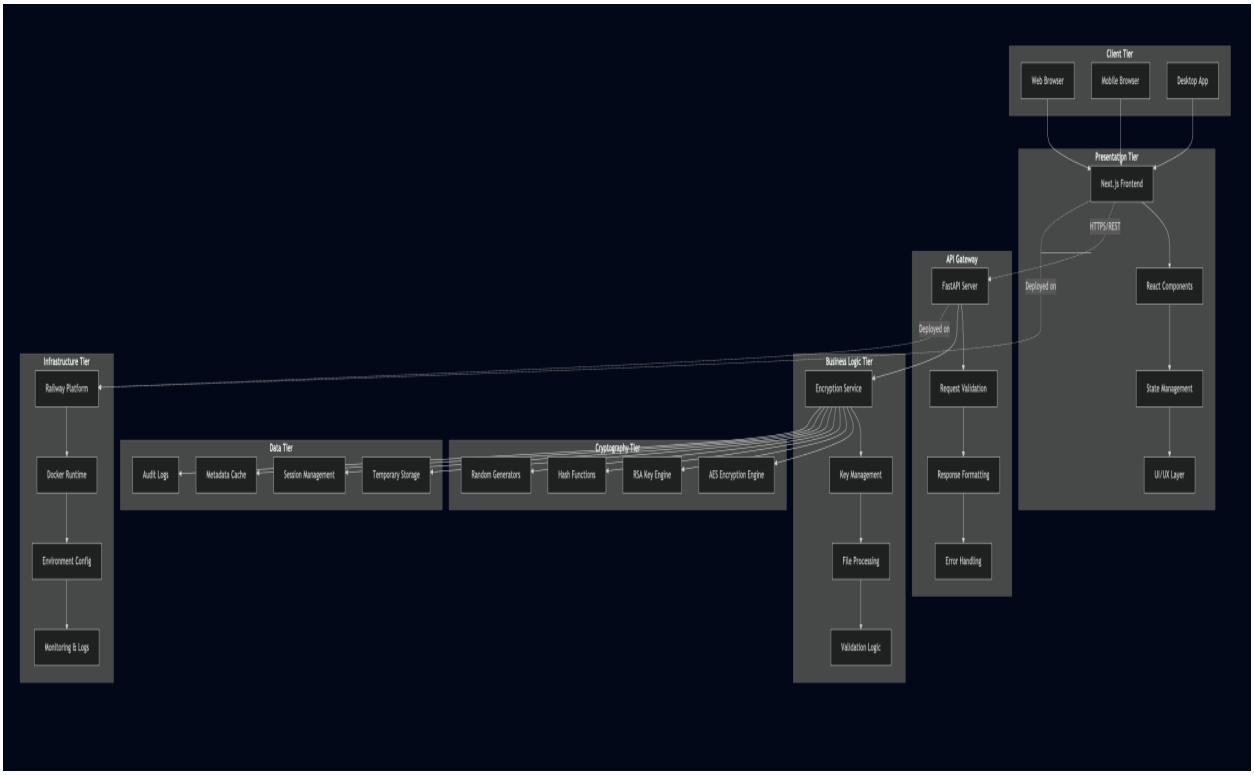
- **Industry Standard**: Python cryptography library with FIPS compliance.
- **Hardware Acceleration**: Optimized implementations using hardware acceleration.
- **Constant-Time Operations**: Resistance to timing attacks.
- **Memory Safety**: Secure memory handling and cleanup.
- **Regular Updates**: Active maintenance and security updates.

3.3.3 Infrastructure Technologies

Railway Deployment Platform

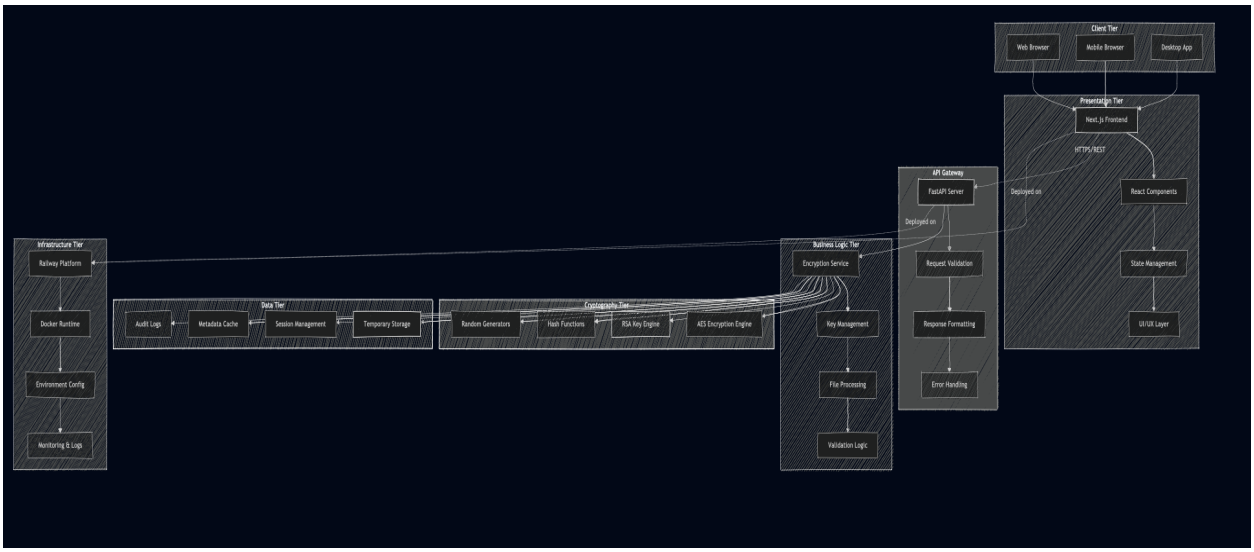
- **Container Orchestration**: Docker-based deployment with automatic scaling.
- **CI/CD Pipeline**: Integrated continuous integration and deployment.
- **Environment Management**: Secure environment variable management.
- **Monitoring**: Built-in application monitoring and logging.
- **Global CDN**: Worldwide content delivery network for optimal performance.

3.4. Block Diagram

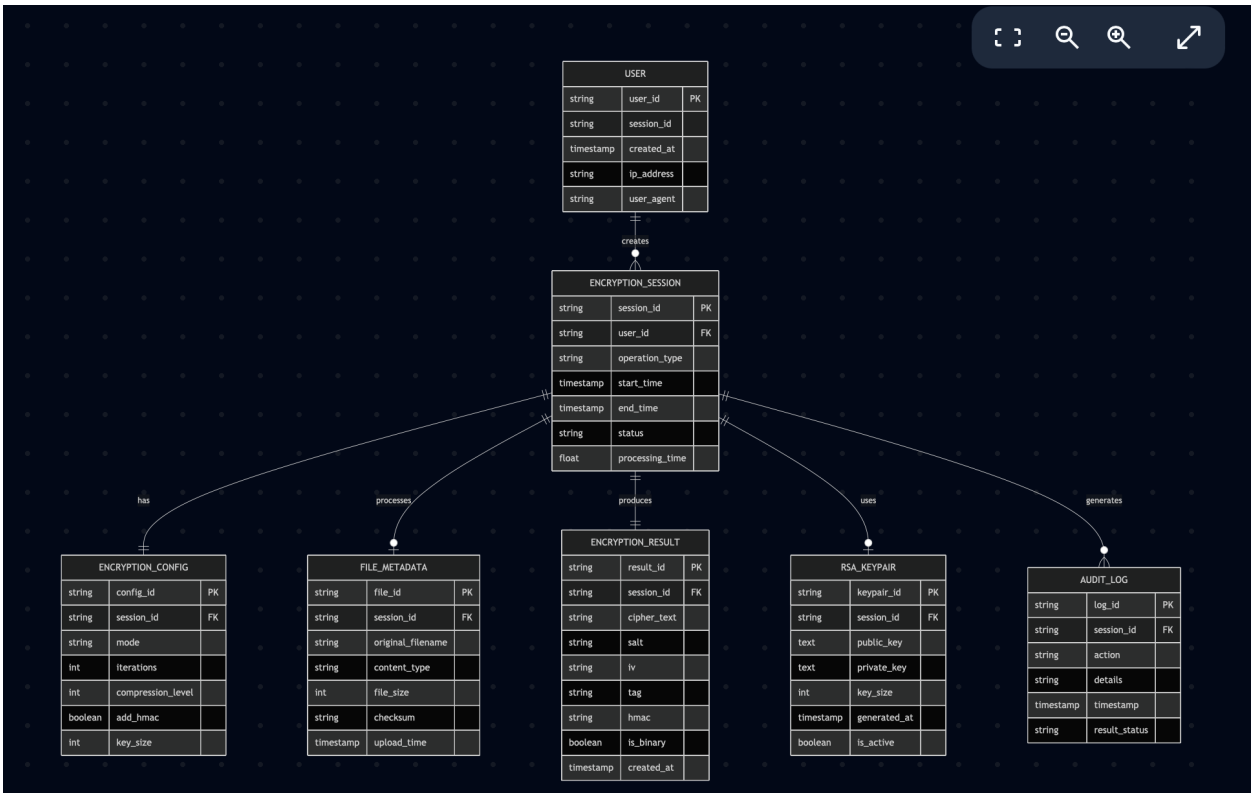


4. Software Design Description

4.1. Design Overview

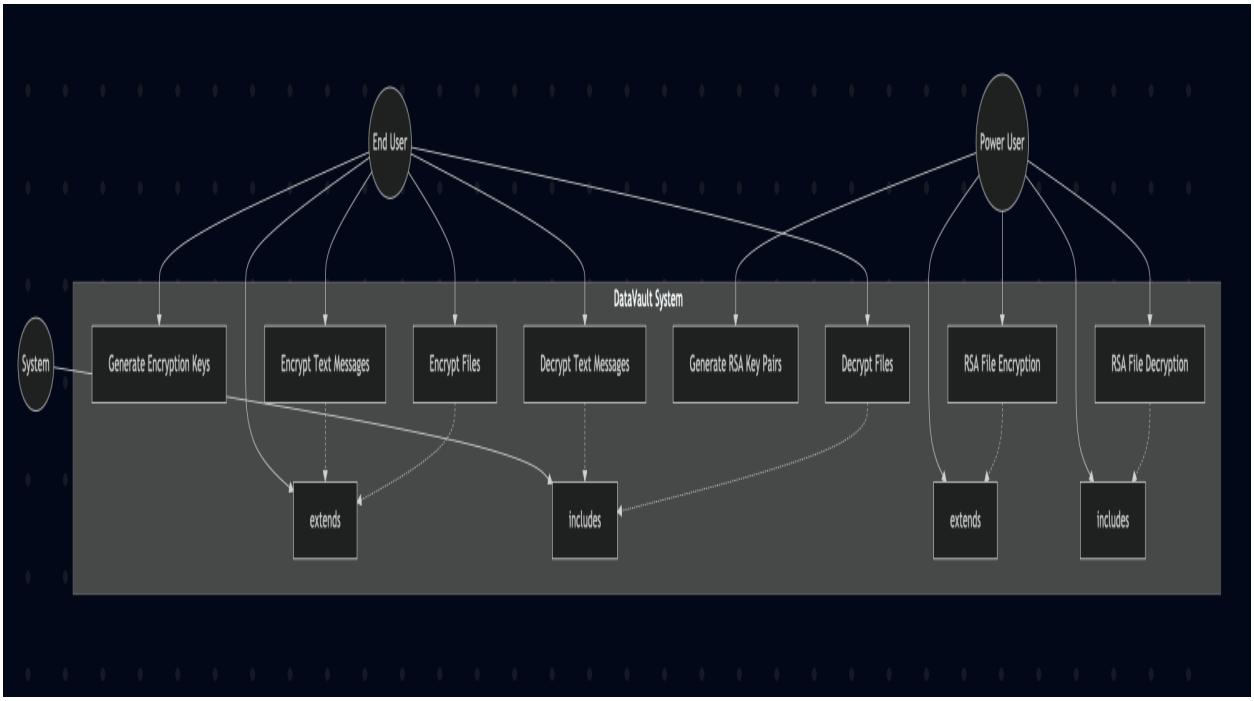


4.2. ER Diagram

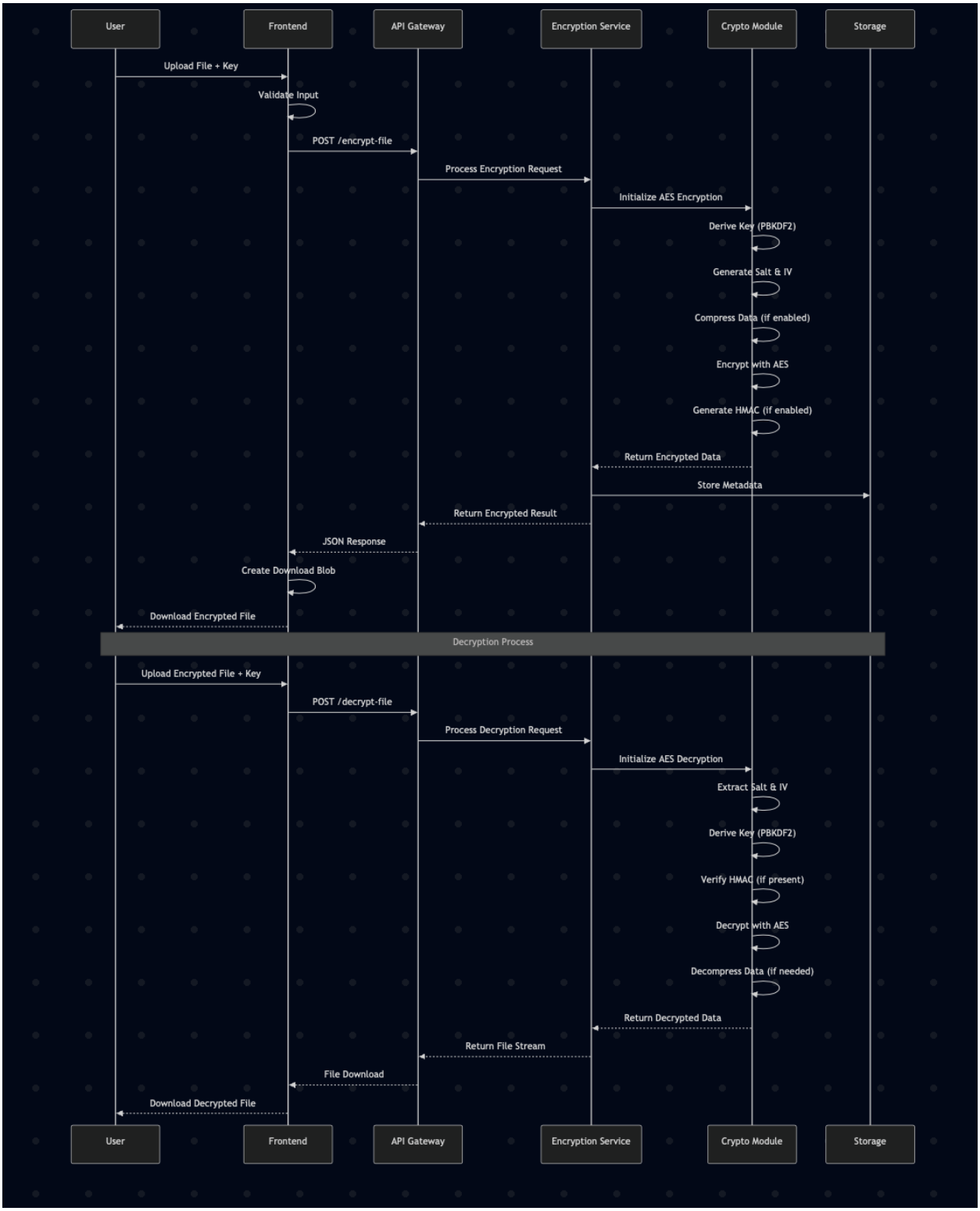


4.3. Work Flow Diagram

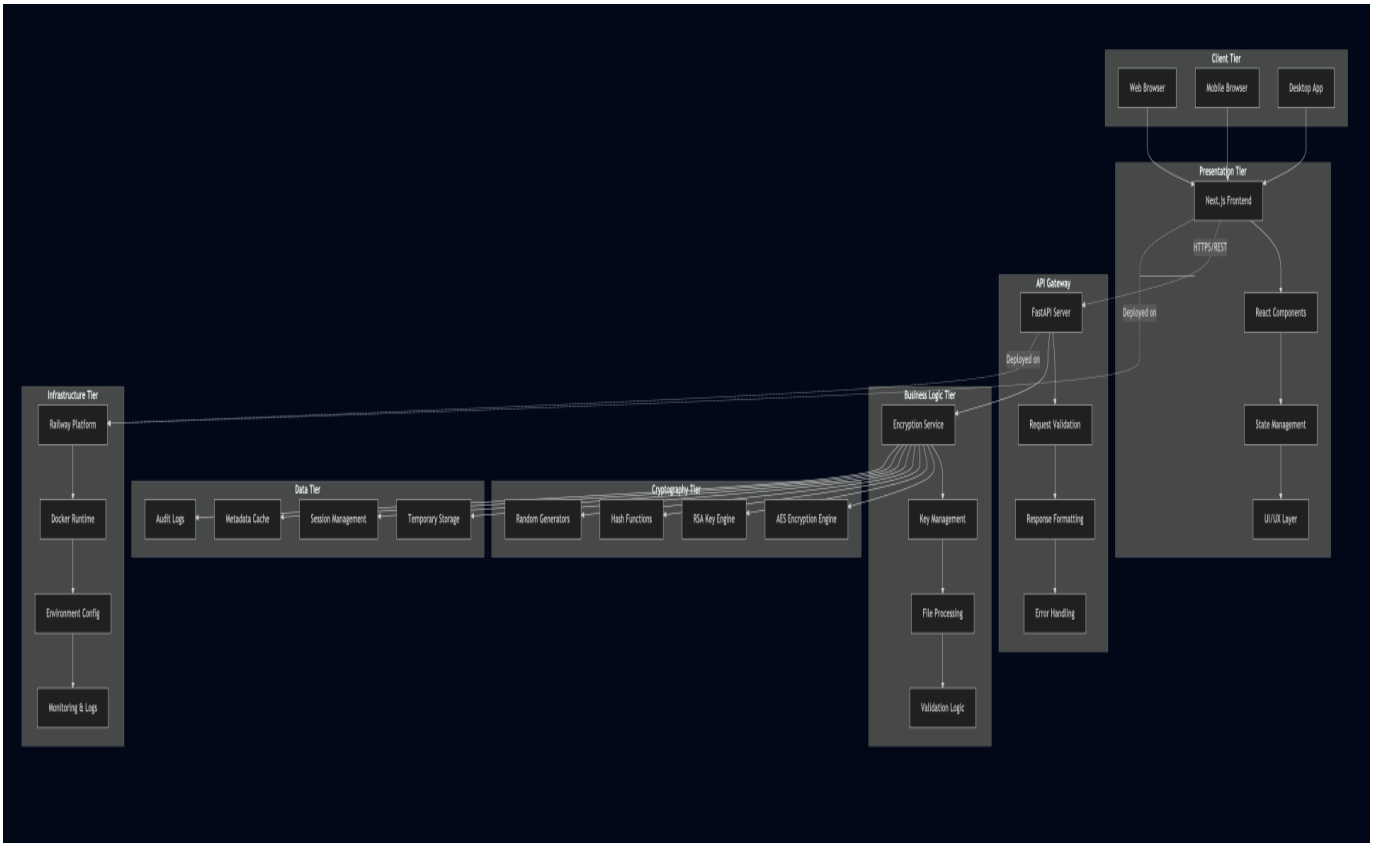
4.4. Use Case Diagram



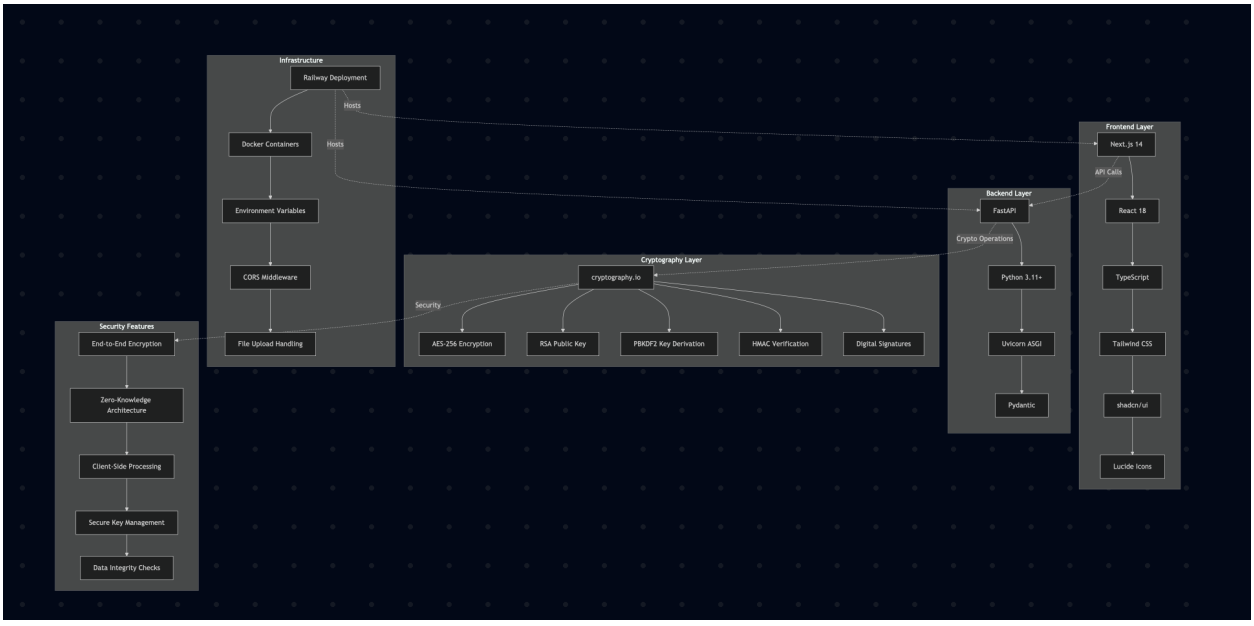
4.5. Sequence Diagram



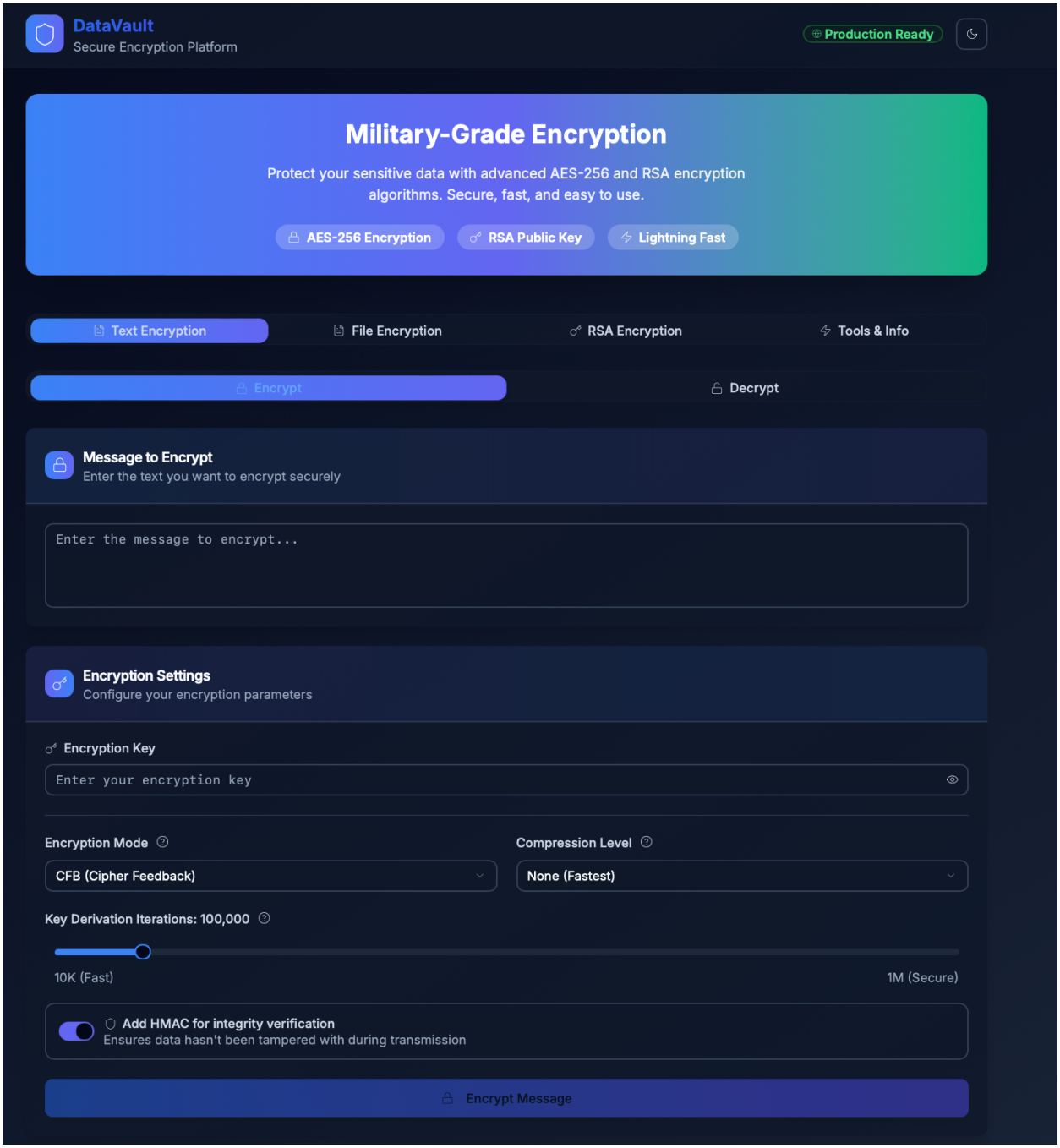
4.6. Block Diagram

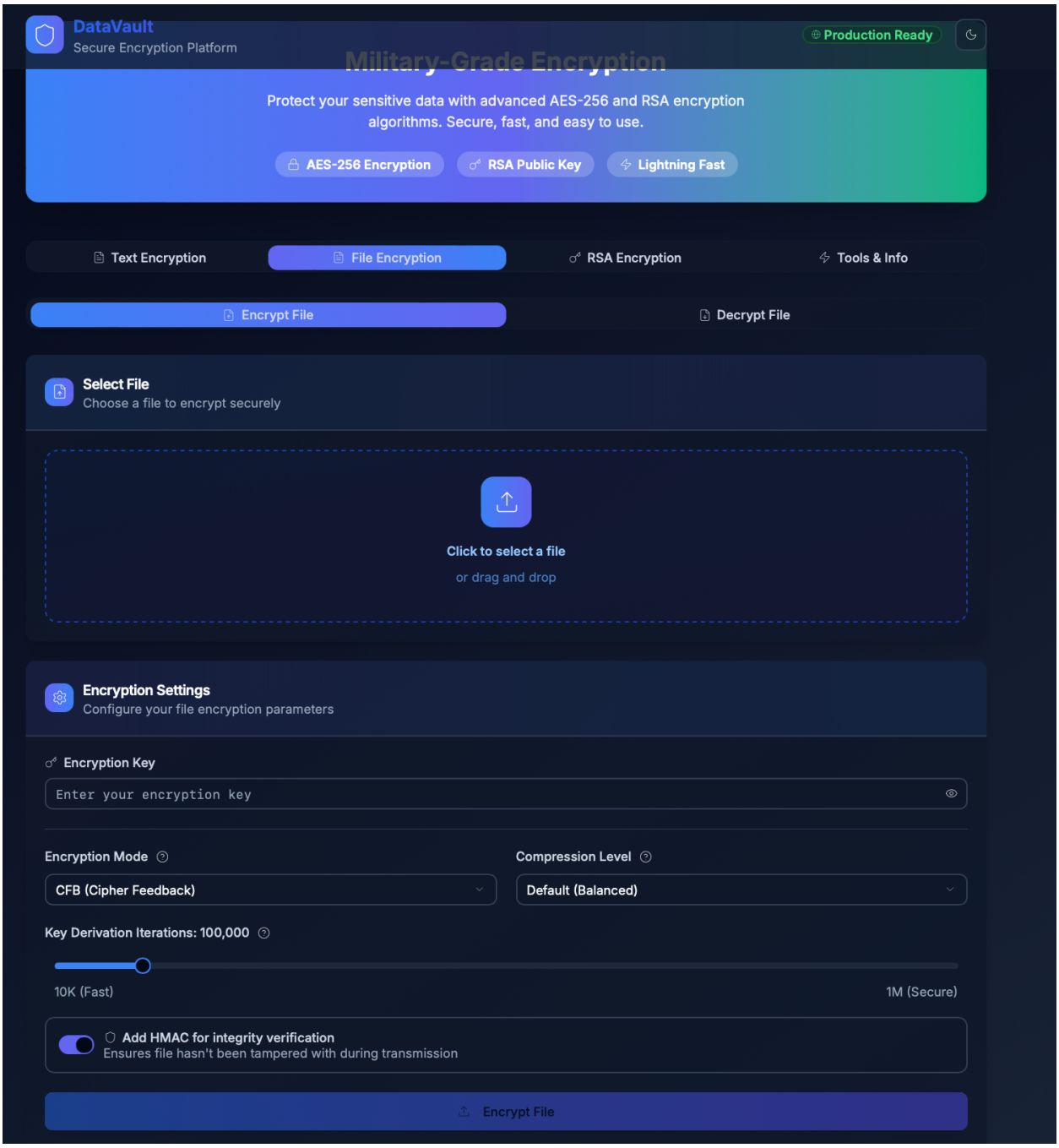


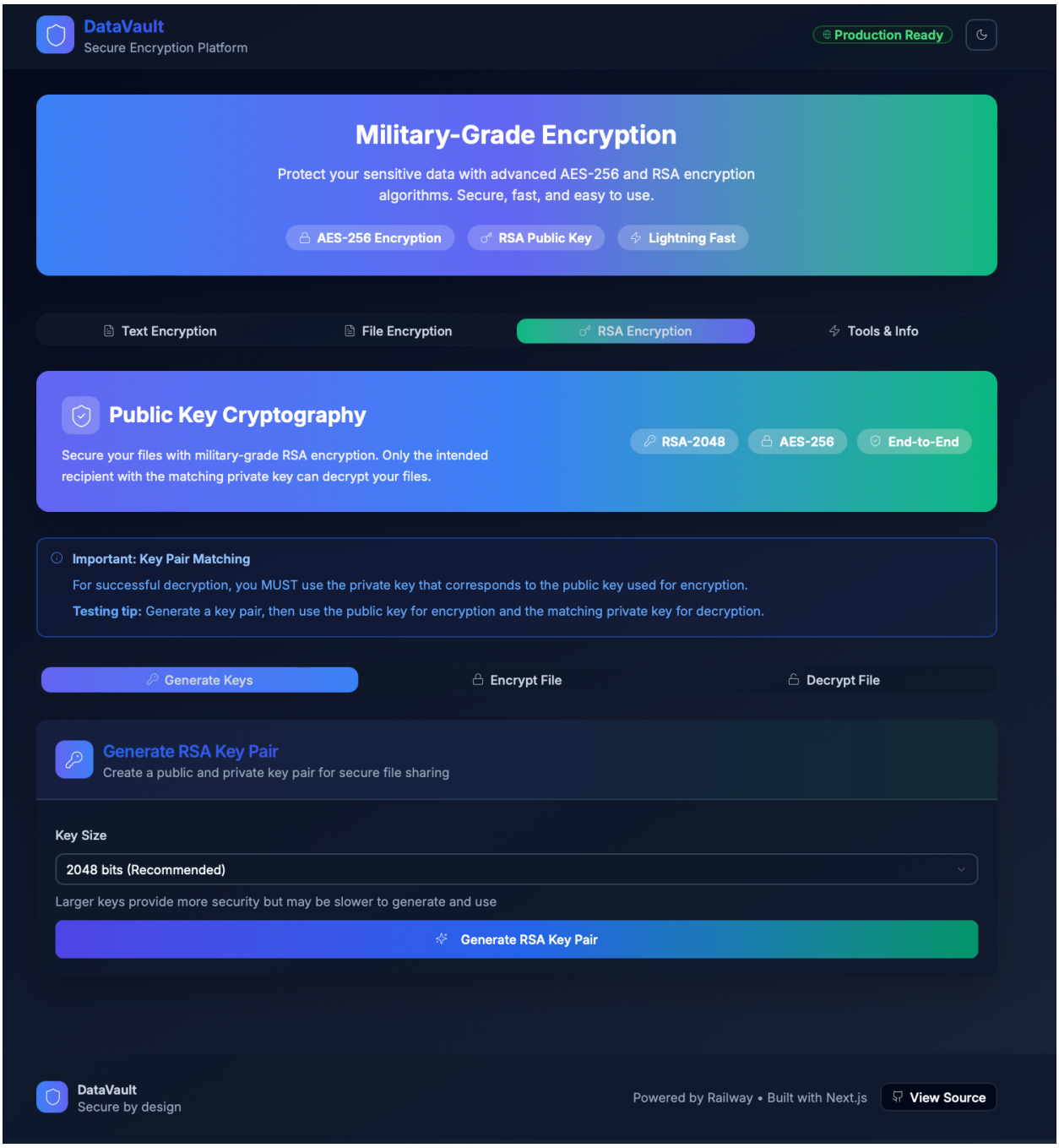
4.7. Technologies being used

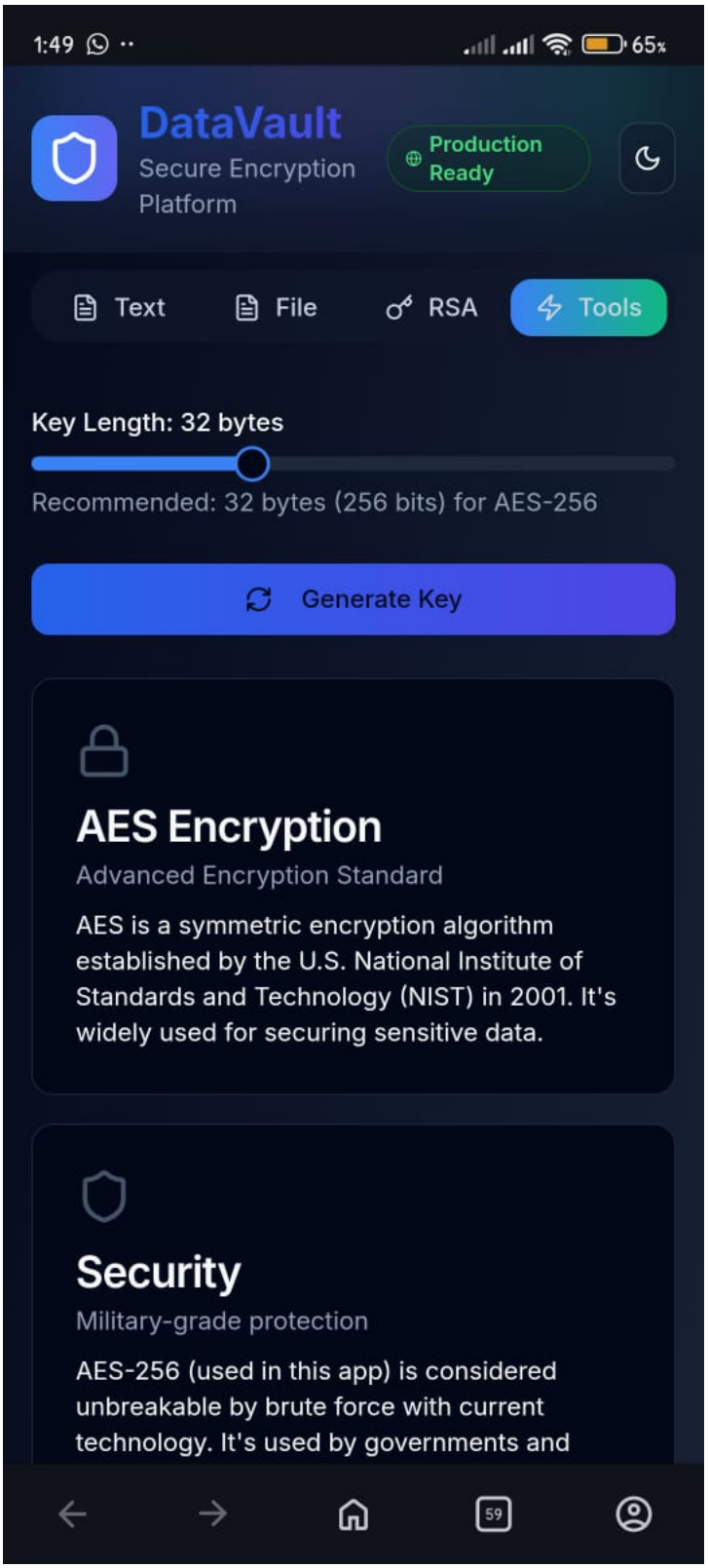


4.8. Screen Images




DataVault
Secure Encryption Platform
Production Ready
Military-Grade Encryption
Protect your sensitive data with advanced AES-256 and RSA encryption algorithms. Secure, fast, and easy to use.
AES-256 Encryption
RSA Public Key
Lightning Fast
Text Encryption
File Encryption
RSA Encryption
Tools & Info
Encrypt File
Decrypt File
Select File
Choose a file to encrypt securely
Click to select a file
or drag and drop
Encryption Settings
Configure your file encryption parameters
Encryption Key
Enter your encryption key
Encryption Mode
CFB (Cipher Feedback)
Compression Level
Default (Balanced)
Key Derivation Iterations: 100,000
10K (Fast)
1M (Secure)
Add HMAC for integrity verification
Ensures file hasn't been tampered with during transmission
Encrypt File


DataVault
Secure Encryption Platform
Production Ready
Military-Grade Encryption
Protect your sensitive data with advanced AES-256 and RSA encryption algorithms. Secure, fast, and easy to use.
AES-256 Encryption
RSA Public Key
Lightning Fast
Text Encryption
File Encryption
RSA Encryption
Tools & Info
Public Key Cryptography
Secure your files with military-grade RSA encryption. Only the intended recipient with the matching private key can decrypt your files.
RSA-2048
AES-256
End-to-End
Important: Key Pair Matching
For successful decryption, you MUST use the private key that corresponds to the public key used for encryption.
Testing tip: Generate a key pair, then use the public key for encryption and the matching private key for decryption.
Generate Keys
Encrypt File
Decrypt File
Generate RSA Key Pair
Create a public and private key pair for secure file sharing
Key Size
2048 bits (Recommended)
Larger keys provide more security but may be slower to generate and use
Generate RSA Key Pair
DataVault
Secure by design
Powered by Railway • Built with Next.js
View Source


1:49
65%
DataVault
Secure Encryption Platform
Production Ready
Text
File
RSA
Tools
Key Length: 32 bytes
Recommended: 32 bytes (256 bits) for AES-256
Generate Key
AES Encryption
Advanced Encryption Standard
AES is a symmetric encryption algorithm established by the U.S. National Institute of Standards and Technology (NIST) in 2001. It's widely used for securing sensitive data.
Security
Military-grade protection
AES-256 (used in this app) is considered unbreakable by brute force with current technology. It's used by governments and
59

5. Project Scope

Functional Scope

5.1.1 Core Functionality

Text Encryption Services

- Encrypt and decrypt text messages using AES-256.
- Support for multiple encryption modes (CFB, CBC, GCM).
- Configurable key derivation parameters.
- Optional HMAC for integrity verification.
- Real-time encryption with immediate results.
- Copy-to-clipboard functionality for encrypted text.
- Support for large text documents up to 10MB.

File Encryption Services

- Encrypt and decrypt files of any type and size (up to 100MB).
- Preserve original file metadata and structure.
- Support for batch file processing.
- Automatic file type detection and handling.
- Secure temporary file management.
- Download encrypted files in standard formats.
- Integrity verification with checksums.

RSA Public Key Cryptography

- Generate RSA key pairs (2048, 3072, 4096 bits).
- Hybrid encryption combining RSA and AES.
- Digital signature creation and verification.
- Secure key exchange mechanisms.
- Public key import/export functionality.
- Key pair management and storage.
- Certificate-like functionality for identity verification.

Key Management System

- Secure random key generation.
- Configurable key length (16-64 bytes).
- Key strength validation and recommendations.
- Key backup and recovery options.
- Key rotation and lifecycle management.
- Secure key storage recommendations.
- Key sharing protocols and best practices.

5.1.2 User Interface Features

Responsive Web Application

- Cross-platform compatibility (Windows, macOS, Linux, mobile).
- Progressive Web App (PWA) capabilities.
- Offline functionality for basic operations.
- Touch-friendly interface for mobile devices.
- Keyboard shortcuts for power users.
- Accessibility compliance (WCAG 2.1 AA).
- Multi-language support framework.

Advanced User Experience

- Drag-and-drop file upload with visual feedback.
- Real-time progress indicators for long operations.
- Dark/light theme switching with system preference detection.
- Customizable user preferences and settings.
- Recent operations history and quick access.
- Export/import of user configurations.
- Contextual help and documentation integration.

5.1.3 Security Features

Advanced Security Measures

- Timing attack resistance in cryptographic operations.
- Side-channel attack mitigation.
- Secure key derivation with configurable iterations.
- Forward secrecy for all encryption operations.
- Tamper detection and integrity verification.
- Secure session management.
- Rate limiting and abuse prevention.

5.1.4 Performance Requirements

Response Time Targets

- Text encryption: < 100ms for messages up to 1MB.
- File encryption: < 5 seconds for files up to 10MB.
- Key generation: < 2 seconds for 4096-bit RSA keys.
- Page load time: < 3 seconds on 3G connection.
- API response time: < 500ms for 95% of requests.
- Database query time: < 100ms average.
- CDN cache hit ratio: > 95% for static assets.

Project Limitations

5.2 Technical Limitations

File Size Constraints

- Maximum file size: 100MB per upload.
- Browser memory limitations for large files.
- Network timeout considerations for slow connections.
- Storage limitations on client devices.
- Processing time constraints for user experience.
- Bandwidth considerations for mobile users.
- Browser compatibility limitations for advanced features.

Cryptographic Limitations

- Quantum computing resistance not guaranteed.
- Key length limitations based on browser capabilities.
- Performance trade-offs for maximum security.
- Compatibility with legacy systems.
- Export restrictions for certain countries.
- Patent considerations for some algorithms.
- Hardware security module integration complexity.

5.2.1 Operational Limitations

Scalability Constraints

- Initial deployment limited to single region.
- Database scaling limitations.
- CDN coverage limitations.
- Support team availability constraints.
- Documentation and training limitations.
- Budget constraints for infrastructure.
- Third-party service dependencies.

6. Module Distribution

This module distribution plan outlines the responsibilities on the DataVault encryption platform. The distribution focuses on frontend, backend, infrastructure, and testing/documentation modules, detailing the tasks each module accomplish, ensuring a balanced workload and efficient collaboration.

Module	Team Member	Responsibilities and Tasks
Frontend	Syed Ahnaf Raza and Memoona Iqbal	- Develop a responsive user interface with a utility-first CSS framework for consistent styling. - Implement client-side encryption using AES-256 and RSA algorithms via the Web Cryptography API. - Create interfaces for text encryption, file encryption, and key generation with intuitive controls. - Manage application state using React hooks and context for seamless user experience. - Ensure accessibility compliance (WCAG 2.1 AA) with screen reader support and keyboard navigation. - Optimize UI for mobile and desktop devices with dark/light theme support. - Handle file uploads/downloads with drag-and-drop functionality and real-time progress indicators.
Backend	Syed Ahnaf Raza and Seemab Asghar	- Build RESTful APIs using a high-performance Python framework for encryption and key management. - Implement server-side cryptographic operations, including AES-256 and RSA key generation. - Develop secure key derivation using PBKDF2 with configurable iterations. - Create endpoints for file handling, encryption operations, and key management. - Ensure secure random number generation and hash functions (SHA-256, HMAC) for integrity. - Validate API inputs/outputs using type-safe models to prevent errors. - Configure middleware for CORS, rate limiting, and security headers to protect the API.

Infrastructure	Syed Ahnaf Raza	- Set up containerized environments using Docker for consistent frontend and backend deployment. - Configure a deployment platform (e.g., Railway) for auto-scaling and global content delivery. - Implement CI/CD pipelines to automate building, testing, and deployment processes. - Set up monitoring tools to track application performance and errors in production. - Configure alerting systems for real-time notifications of system issues. - Ensure secure environment variable management and load balancing for scalability. - Optimize infrastructure for 99.9% uptime with automated failover capabilities.
Testing & Documentation	Izaan Abdullah and Memoona Iqbal	- Write unit, integration, and end-to-end tests for frontend and backend components. - Perform security testing to validate cryptographic implementations and detect vulnerabilities. - Ensure 90% test coverage for critical encryption and API functions. - Create user guides and FAQs to assist non-technical users with encryption tasks. - Develop technical documentation covering API endpoints, security model, and setup guides. - Document threat models and security policies to ensure compliance with GDPR. - Conduct accessibility and performance testing to verify UI responsiveness and compliance.

7. Code (Complete Backend live on Railway)

https://aes-rsa-production.up.railway.app

https://github.com/syahra712/AES-RSA

https://v0-data-vault-drab.vercel.app

8. Conclusion

DataVault represents a comprehensive solution to the growing need for accessible, secure, and user-friendly encryption tools in today's digital landscape. By combining military-grade cryptographic algorithms with modern web technologies, the platform successfully bridges the gap between security and usability that has long plagued the encryption software market.

9. References

- **"2023 Data Breach Investigations Report"** - Verizon Business
Verizon Communications Inc.
https://www.verizon.com/business/resources/reports/dbir/
- **"Cost of a Data Breach Report 2023"** - IBM Security
International Business Machines Corporation
https://www.ibm.com/security/data-breach
- **"Cybersecurity Framework Implementation Guidance"** - NIST
National Institute of Standards and Technology
https://www.nist.gov/cyberframework/implementation-guidance
- **"Web Application Security Testing Guide"** - OWASP
Open Web Application Security Project
https://owasp.org/www-project-web-security-testing-guide/
- **NIST FIPS 197** - Advanced Encryption Standard (AES)
National Institute of Standards and Technology
https://csrc.nist.gov/publications/detail/fips/197/final
- **RFC 3447** - Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version
Internet Engineering Task Force (IETF)
https://tools.ietf.org/html/rfc3447
- **RFC 2898** - PKCS #5: Password-Based Cryptography Specification Version 2.0
Internet Engineering Task Force (IETF)
https://tools.ietf.org/html/rfc2898
- **NIST SP 800-38A** - Recommendation for Block Cipher Modes of Operation
National Institute of Standards and Technology
https://csrc.nist.gov/publications/detail/sp/800-38a/final
- **NIST SP 800-38D** - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM)
National Institute of Standards and Technology
https://csrc.nist.gov/publications/detail/sp/800-38d/final