Project Ragnarok:
Post-Quantum Cybersecurity with Lightning data

Design Document

Group Members:

Joanna Zhang

Gianni Bubb

Aidan Nelappana

Faculty Advisor/Client:

Dr. Bhattacharyya

Florida Institute of Technology Senior Design

Table of Contents

1. Introduction

1.1 Introduction

This design document describes the architecture and detailed design of **Project Ragnarok**, a post-quantum database security system that incorporates lightning data as a high-entropy source for cryptographic operations. The document defines how the system's components, modules, and algorithms are structured to satisfy the functional and non-functional requirements outlined in the corresponding requirements specification.

1.2 Purpose

The purpose of this document is to present the technical design of Project Ragnarok, including its system architecture, modular components, and cryptographic workflows. It serves as a blueprint for implementation and provides detailed descriptions of how the system satisfies the specifications defined in the Project Ragnarok Requirements Document.

1.3 Intended Audience

This document is intended for project developers, faculty advisors, and technical stakeholders involved in the design, implementation, and evaluation of the system. It assumes a foundational understanding of database systems, cybersecurity principles, and cryptographic concepts.

1.4 References

The following documents and standards provide additional context and guidance relevant to this document:

- Project Ragnarok Requirements Document: https://prismaticr.github.io/docs/milestone1/Milestone_1_Requirements.pdf
- FIPS 203: https://csrc.nist.gov/pubs/fips/203/final
- FIPS 204: https://csrc.nist.gov/pubs/fips/204/final
- FIPS 205: https://csrc.nist.gov/pubs/fips/205/final

2. System Design

2.1 Design Overview

The goal of our application is to provide new methods of post-quantum cryptography and keys based on better random numbers. We plan to accomplish this by giving the user access to our methods through an API. The system follows a modular, layered architecture separating user interaction, cryptographic services, and database management. This design improves security isolation, supports cryptographic agility, and allows independent updates to post-quantum algorithms.

2.2 System Architecture

This section describes the overall architectural structure of Project Ragnarok and the organization of its primary components. The system adopts a layered architecture to separate user interaction, cryptographic processing, entropy management, and data storage into distinct functional areas. This separation enhances security isolation, reduces system complexity, and supports modular development. The architecture defines how user-facing components communicate with internal services and how lightning-derived entropy flows through the cryptographic engine to protect stored data.

2.2.1 User Types

Project Ragnarok defines three primary user roles: Administrators, Authorized Database Clients, and Security Auditors. Administrators manage system configurations, oversee cryptographic policies, and monitor system activity. Authorized Database Clients access API services to generate keys and perform encryption or decryption operations. Security Auditors review logs and verify compliance with security policies. Role-based access control ensures that each user type operates with appropriate permissions.

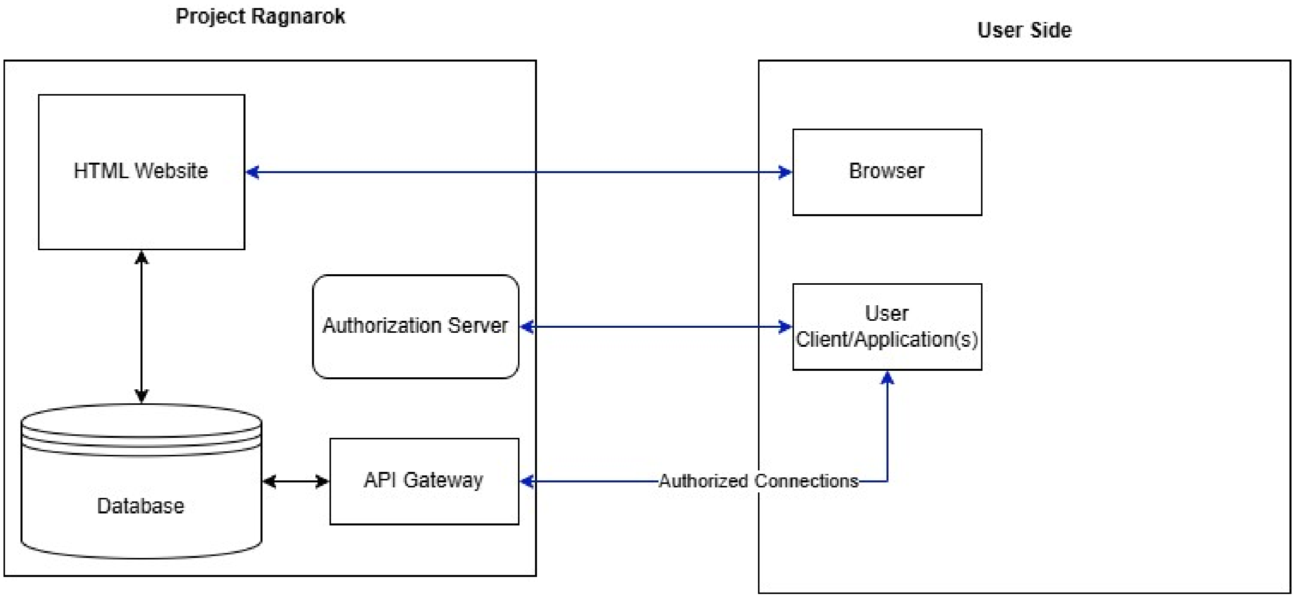
2.2.2 User-Facing Machines

User-facing machines provide the interface between users and the system. These include web-based dashboards, command-line interfaces (CLI), administrative control panels, and secure API endpoints. The API authorization and authentication mechanisms ensure that only verified users can access cryptographic services. All user interactions occur through secure communication channels.

2.2.3 Non-User-Facing Machines

Non-user-facing machines perform the system's internal processing and are isolated from direct external access. These include the Lightning Data Ingestion Service, Entropy Processing and Key Generation Service, Post-Quantum Cryptographic Engine, and Secure Database Server. Together, these components handle entropy extraction, key generation, cryptographic operations, and secure data storage while maintaining strict security controls.

2.2.4 Architectural Design Diagram



The architectural design diagram illustrates a two-sided structure consisting of the **User Side** and the **Project Ragnarok System Side**, demonstrating how external users securely interact with internal services.

On the **User Side**, the system includes a web browser and a user client application. The web browser connects directly to the HTML website hosted within Project Ragnarok to access informational pages and user dashboards. The user client application communicates with the Authorization Server to authenticate and obtain access credentials. Once authenticated, the client application establishes authorized connections to the API Gateway to request cryptographic services.

On the **Project Ragnarok System Side**, the HTML website serves as the public-facing interface and connects to the internal database for retrieving or displaying relevant information. The database communicates with the API Gateway to support secure service operations. The API Gateway acts as the central entry point for all cryptographic service requests and enforces security policies. It works in conjunction with the Authorization Server, which validates user credentials, issues authentication tokens, and ensures that only authorized requests are processed.

This two-sided design clearly separates external user interaction from internal system services, ensuring controlled access, secure authentication, and structured communication between presentation components and backend cryptographic infrastructure.

2.3 Module Design

Each module is designed with a single responsibility:

2.3.1 Lightning Data Module

The Lightning Data Module is responsible for collecting, validating, and preprocessing lightning strike data obtained from trusted sources. This module extracts high-quality entropy from the processed data and prepares it for use in cryptographic operations. By converting environmental lightning activity into usable randomness, the module strengthens the unpredictability of the system's key generation processes.

2.3.2 Post-Quantum Key Generation Module

The Post-Quantum Key Generation Module uses lightning-derived entropy to generate cryptographic keys that are resistant to quantum computing threats. This module ensures that all keys are produced using secure randomness sources and follow post-quantum cryptographic standards. It supports secure key creation and contributes to the overall strength of encryption mechanisms within the system.

2.3.3 Cryptographic Operations Module

The Cryptographic Operations Module performs encryption and decryption of database data using post-quantum algorithms. It protects sensitive information during storage (data-at-rest) and transmission (data-in-transit). This module serves as the core processing component that applies cryptographic protections to system data.

2.3.4 Secure Database Module

The Secure Database Module manages persistent storage within the system. It stores encrypted data exclusively and enforces strict access control policies to prevent unauthorized access. This module also supports audit logging and ensures that sensitive information remains protected throughout its lifecycle.

3. Post-Quantum Cryptographic Algorithm Design

The system employs post-quantum cryptographic primitives for key exchange, encryption, and authentication. Lightning-derived high-entropy randomness is used to seed cryptographic processes (increasing resistance to predictability and entropy-based attacks). Algorithm selection and integration are designed to be modular, allowing future replacement as post-quantum standards evolve. This modular design enables cryptographic agility (so that new PQC algorithms can be incorporated without major architectural changes).

For implementation, the system uses the **liboqs** library (an open-source C library for quantum-safe cryptographic algorithms maintained by the Open Quantum Safe project) and its associated provider integrations (such as an OpenSSL PQC provider) to access a broad suite of post-quantum key encapsulation mechanisms (KEMs) and signature schemes. liboqs supports several algorithm families, including those standardized or selected by **NIST in its post-quantum cryptography standardization process** (such as the Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM, formerly CRYSTALS-Kyber), Module-Lattice-Based Digital Signature Algorithm (ML-DSA, formerly CRYSTALS-Dilithium), and Stateless Hash-Based Signature schemes (SLH-DSA, formerly SPHINCS+), with additional candidates such as HQC and others available for experimentation) (see NIST PQC standards and algorithm lists).

By aligning with these NIST-recognized post-quantum algorithms and implementing them through a well-maintained open-source library, the system prepares cryptographic operations for resistance against both classical and emerging quantum adversaries. The design anticipates future standards and algorithm enhancements, supporting key generation, encryption, and digital signature operations that conform to evolving FIPS and NIST recommendations (such as FIPS 203, FIPS 204, and FIPS 205 for PQC key encapsulation and signatures).

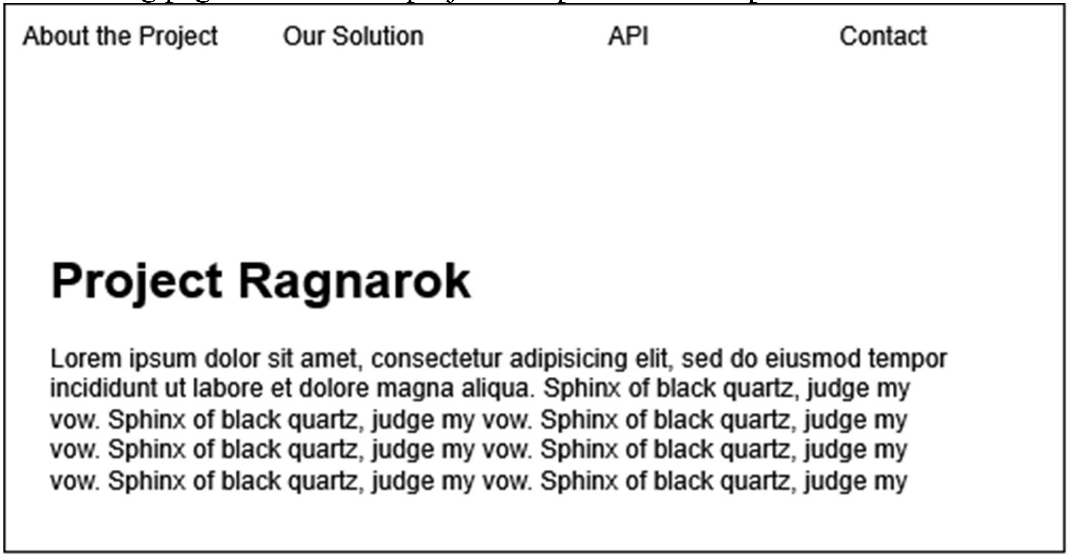
4. Web Application Interface Design

We want to provide a similar web page to Project Thor which describes the problem, the project and its purpose, our solution, an “about the team" page, a contact page, and a dedicated API page.

4.1 Page Descriptions

4.1.1 Landing Page

The landing page will have the project and problem description.



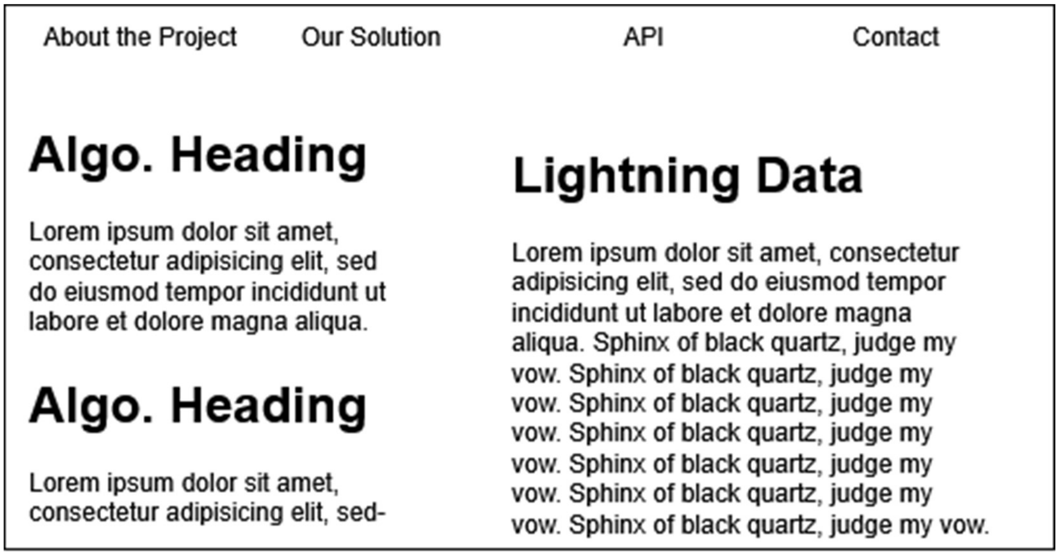
Project Ragnarok

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my

Prototype Images Subject to Change

4.1.2 Solution Page

The solution page will have a more detailed description on how we generate keys and our algorithms.



Algo. Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Algo. Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed-

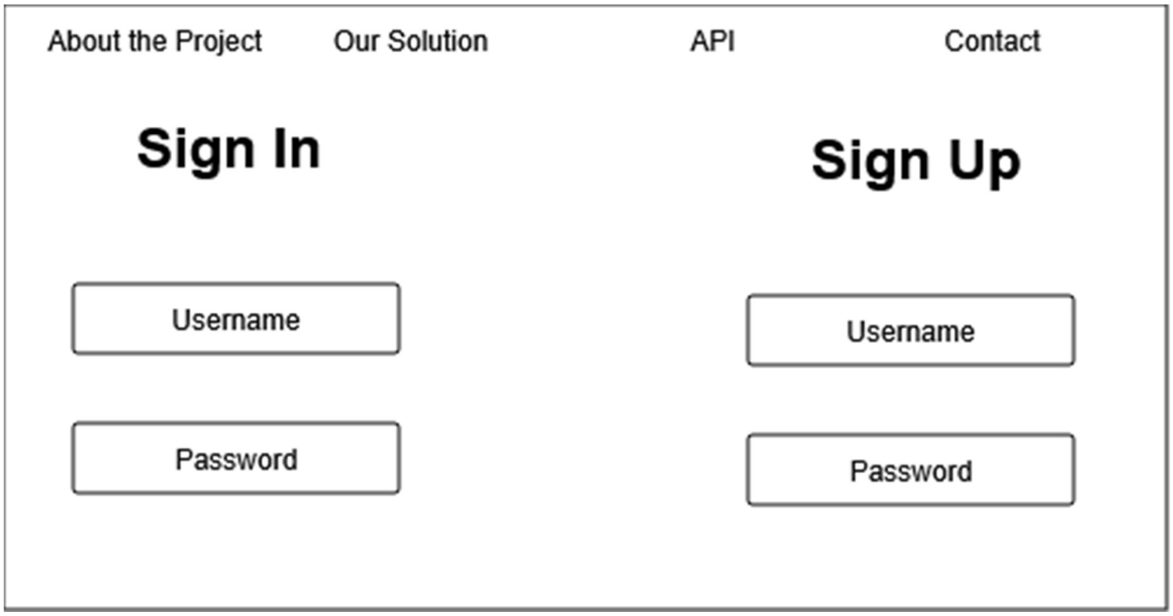
Lightning Data

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow. Sphinx of black quartz, judge my vow.

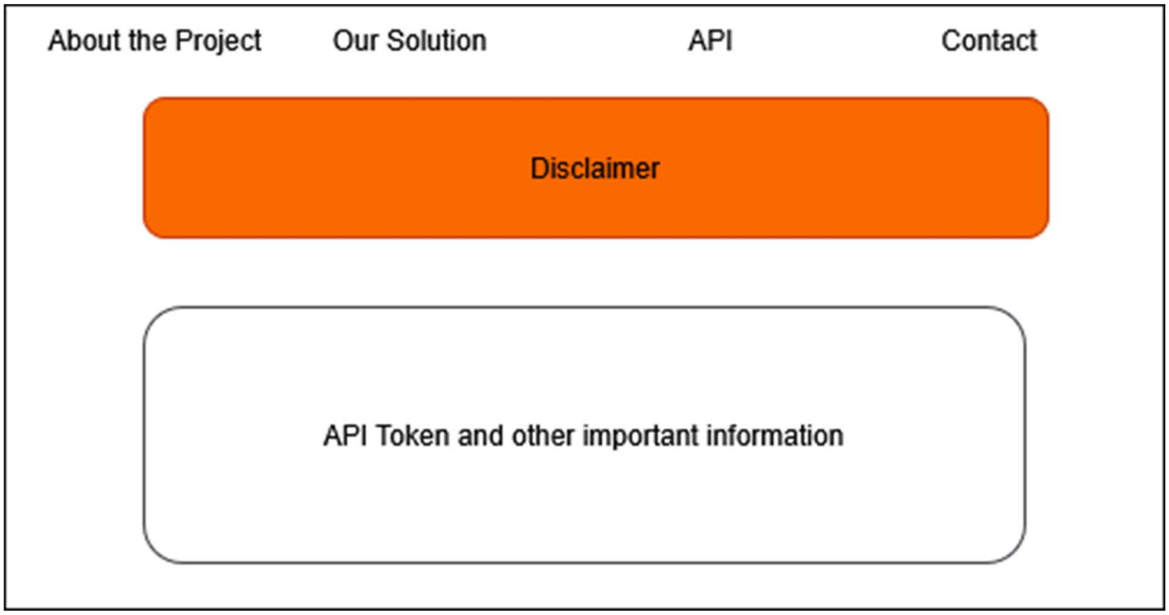
Prototype Images Subject to Change

4.1.3 API Page

The API page will have a dashboard which will let users sign up for API tokens. After users successfully login and apply for a token, they will be shown their perishable token once. This page will include a disclaimer describing this and other important information as we see fit.

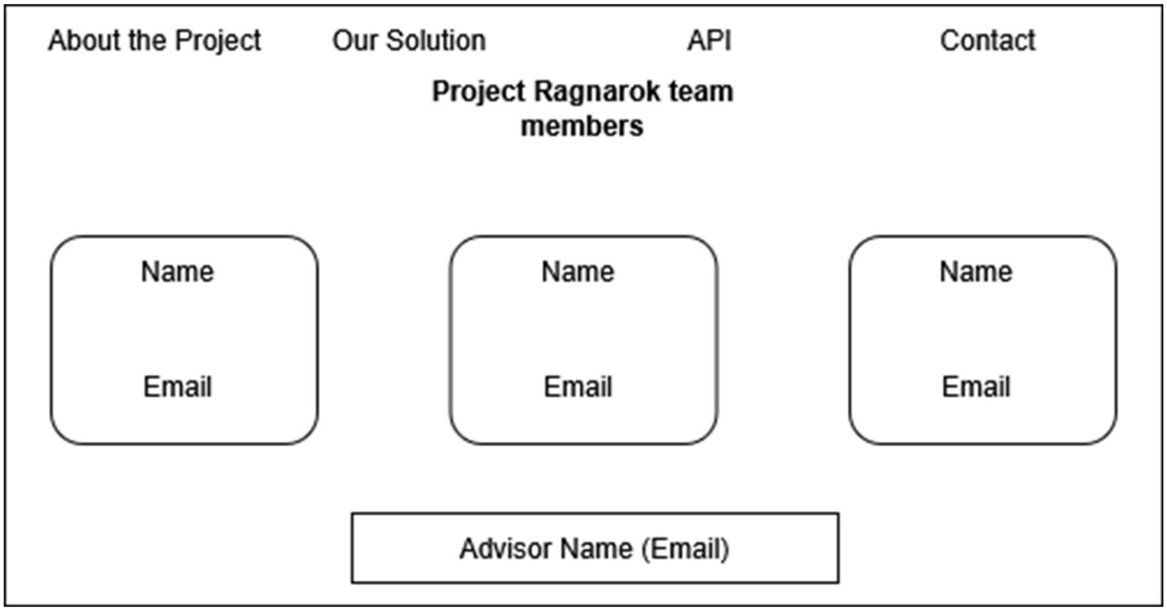


Prototype Images Subject to Change



4.1.4 Contact Page

The contact page will list all the team members and their respective email addresses. Users will be able to copy our emails.


About the Project
Our Solution
API
Contact
Project Ragnarok team members
Name
Email
Name
Email
Name
Email
Advisor Name (Email)