Architecture Plan

Tyler D. Wallingford

CST-452 Senior Project II

Grand Canyon University

Instructor: Professor Jeff Stucker

Date: 09/29/24

**Architecture Plan**

**1. Overview**

The Secure Messaging Application is designed to provide a secure and encrypted communication platform for users, ensuring data confidentiality, integrity, and availability. The application employs end-to-end encryption using AES-128 and adheres to security best practices, including CSRF protection, CSP implementation, and deployment on Microsoft Azure.

**2. Architectural Components**

**a. Application Layer**

* **Frontend**: HTML, CSS (with light/dark themes and style), JavaScript
  + The UI allows users to send and receive messages, add contacts, and manage their account settings.
  + Interacts with the backend using AJAX requests and fetch APIs.
* **Backend**: PHP
  + Handles user authentication, message processing, encryption/decryption, contact management, and data retrieval.

**b. Security Layer**

* **AES-128 Encryption**: Ensures end-to-end encryption for messages, with encryption and decryption happening on the backend.
* **CSRF Protection**: Implemented via CSRF tokens to prevent unauthorized form submissions.
* **Content Security Policy (CSP)**: Protects against XSS and data injection attacks by specifying allowed content sources.

**c. Data Layer**

* **Database**: MySQL hosted on Microsoft Azure
  + Stores user information, messages, contacts, and encryption keys.

**d. Infrastructure Layer**

* **Cloud Hosting**: Microsoft Azure
  + PHP application hosted on an Azure Web App.
  + Database managed using Azure Database for MySQL.
  + Load balancing and auto-scaling features provided by Azure ensure high availability and fault tolerance.

**3. Detailed Architecture**

**3.1. User Registration and Authentication**

* Users register and log in to the application, where passwords are hashed using a secure algorithm (bcrypt) before storage.
* CSRF tokens are generated upon login and stored in the session.

**3.2. Message Flow**

* When a user sends a message, it is encrypted using AES-128 before being sent to the server.
* The encrypted message is stored in the MySQL database and retrieved using secure queries.
* Messages are decrypted on the backend before being displayed to the recipient.

**3.3. Contact Management**

* Users can add and manage contacts, with usernames stored in the database.
* CSRF protection ensures that unauthorized requests cannot alter contact information.

**4. Deployment Plan**

**4.1. Microsoft Azure Setup**

* **Azure Web App**: The PHP application is deployed here.
* **Azure Database for MySQL**: The database is set up and connected to the web application.
* **Azure Storage Account**: Used for data scalability.

**5. Security Measures**

* **Encryption**: AES-128 encryption for messages.
* **CSRF Protection**: Implemented via CSRF tokens.
* **CSP Headers**: Ensuring only trusted sources are allowed for scripts, styles, and other content.
* **Htmlspecialchars()**: A security feature within PHP used to mitigate XSS by escaping special HTML characters which prevents malicious code injections.

**6. Monitoring and Maintenance**

* **Logging**: Application errors and access logs are monitored using Azure Monitor.
* **Alerts**: Set up on Azure to notify the system administrator of any potential security breaches or service downtimes.