CSE 345/545 Foundations to Computer Security

Course Project Requirements

Social Media Marketplace

1. Introduction

The **Social Media Platform** is intended to provide **end-to-end security** for group interactions, private messages, media sharing, and P2P Marketplace. The goal is to build a robust system that ensures **confidentiality**, **integrity**, and **availability** of all user communications—while integrating **user validation mechanisms**, **OTP-based authentication**, and **PKI** for secure operations.

2. Requirements

Below are the key requirements, reimagined for a messaging and media-sharing application:

1. **End-to-End Encrypted Conversations**
    - Direct messaging (one-to-one).
    - Group messaging (many-to-many).
    - Optional ephemeral (disappearing) messages or stories.
2. **Secure Media-Sharing**
    - Users can share photos, videos, voice notes, or documents privately.
    - All shared content must be encrypted in transit (HTTPS/SSL/TLS).
    - The system may optionally implement end-to-end encryption for attachments to provide additional security.
3. **User Identity and Validation**
    - User must verify their email address and mobile number during registration using OTP-based verification
    - User accounts should include fields such as username, profile picture, and optional public bio.
    - The system should flag suspicious activities such as repeated login failures or anomalous behavior for admin review.
4. **Social Features**
    - **Follow** or **Friend Requests**: Users can connect with each other.
    - **Search** for users by username, hashtags, or public profile info.
    - **Block or Report** suspicious accounts or content to the admin.
5. **P2P Marketplace**
    - **Listing of artifacts for sale**
    - **Search functionality**

    - **Payment gateway to facilitate purchases**
6. **Admin & Moderation**
    - An **Admin dashboard** to view and manage all users (verified or unverified).
    - Admin can **suspend or remove** accounts for violating content guidelines.

Additional Functionalities / Security Mandates

- **Public Key Certificates (PKI)**
    - The platform **must use HTTPS** (TLS/SSL) to secure data in transit.
    - At least **two functions** (e.g., account creation, password reset, or certain message-verification steps) must use PKI to ensure authenticity and integrity.
- **OTP with Virtual Keyboard**
    - For **at least two** high-sensitivity transactions or actions (e.g., finalizing account re-verification, password reset, or admin-level actions), require an **OTP** that users must enter through a **virtual keyboard** to mitigate keylogging risks.
- **Secure Logging & Audit**
    - Log all critical actions (user registration, admin moderation, suspicious content flags) in a secure manner (tamper-resistant logs).
- **Defenses Against Attacks**
    - Apply standard security best practices against **SQL injection, XSS, CSRF, session hijacking**, etc.
- **Data Storage Compliance**
    - Do not store **plain-text passwords** or raw **credit card** data (if in-app purchases or premium features are introduced).
    - Use **hashed/salted** passwords and tokenized payment methods if needed.
- **Scalability & Simultaneous Access**
    - Multiple users should be able to exchange messages, upload media, and search the platform concurrently without compromising security.

3. User Roles

1. **Regular Users**
    - **Sign Up / Log In**: Validate account via email and OTP verification.
    - **Messaging**: Send, receive, and manage private/group messages.
    - **Media Sharing**: Upload images/videos in direct or group chats.
    - **Profile Management**: Maintain personal details, handle friend/follow requests.
    - **Report / Block** malicious or spam users.
2. **Admin (Platform Moderators)**
    - **User Management**: View all user accounts, handle suspicious activity.
    - **Moderation**: Remove or suspend abusive/spam accounts based on legal requirements.

- **Verification**: Handle exceptions or manual checks.
- **Security Audits**: Access secure logs, verify system integrity.

4. Programming Languages and Frameworks

- **Operating System**: Ubuntu(will be provided to you)
- **Database**: MySQL, PostgreSQL, MongoDB, or SQLite (others with TA approval).
- **Web Server**: Nginx, Apache, or IIS (others with TA approval).
- **Languages/Frameworks**: Any

5. Milestones & Timeline (January–April)

Your TAs will evaluate your progress in regular check-ins and at designated milestone demos. Below is milestone schedule

---

January Milestone [No Credit] (31 Jan)

1. **Set Up Technology Stack**
   - Choose OS, database, web server, and programming language(s).
   - Configure **HTTPS** with your own certificate authority or a self-signed certificate.
2. **Prototype Deployment**
   - Deploy a simple "Hello World" or skeleton website on the VM / server with **SSL/TLS**.

February Milestone [2.5%] (Feb 28)

1. **Basic User Flows**
   - Implement **User Registration & Login** (with secure password handling).
   - Provide **Profile Management** (update username, profile picture).
2. **Messaging Prototype**
   - Enable **1:1 direct messaging** with a basic UI.
   - Store messages securely in the database (encrypted).
3. **Admin Dashboard (Basic)**
   - View a list of registered users.
   - Manually verify or reject user documents (if not auto-verified).

March Milestone [2.5%] (March 31)

1. **Verification Workflow**
    - Integrate the authentication for real or simulated identity checks.
    - Ensure verified users can access advanced features (e.g., group chats, media sharing).
2. **Group Messaging & Media Sharing**
    - Implement **group chats**.
    - Allow **image/video** uploads in messages.
    - Consider basic **end-to-end encryption** or advanced encryption for attachments.

3. **P2P Marketplace**
    - Payment Gateway Simulations

April Milestone (Final Demo)

1. **OTP & PKI Enhancements**
    - Enable **OTP with virtual keyboard** for at least **two** high-risk actions (e.g., password reset, account closure, admin-level content removal).
    - Integrate **PKI** (public/private key features) into at least **two** functions (e.g., message signing, user verification).
2. **Advanced Moderation & Reporting**
    - Implement robust **report/block** features.
    - Admin can handle violation reports, ban users, or review content if necessary.
3. **Performance & Security Testing**
    - Demonstrate your platform's **defense** against common attacks (SQL injection, XSS, CSRF, session hijacking).
4. **Final Presentation & Documentation**
    - **Demo** the end-to-end system.
    - Provide project documentation, including **architecture diagrams**, **encryption details**, and **security approaches**.

---

6. Bonus Opportunity

+10% BONUS: If your team incorporates **Blockchain** for message integrity (e.g., storing message hashes, user identity verifications, or moderation logs in a private blockchain to ensure immutability), you will be eligible for **up to 10% additional marks**.