Proyecto de Criptografía

Objetivo: Desarrollar una aplicación práctica que utilice **esquemas criptográficos reales** para resolver un problema concreto. El proyecto busca que los estudiantes apliquen los conceptos teóricos vistos en clase en un entorno realista, enfrentándose a decisiones de diseño, seguridad y comunicación. **El proyecto se puede realizar en grupos de hasta 3.**

Opciones de Proyecto

1. **Administrador de Contraseñas Seguro**

- Implementar un sistema que almacene contraseñas de manera cifrada.
- Requisitos mínimos:
    - Aplicación que permita interactuar con contraseñas, crearlas, editarlas, generarlas automáticamente, búsqueda, etc.
    - Servidor que permita guardar blob encriptado, con autenticación de por medio.
    - Derivación de llave a partir de una contraseña maestra (Argon2, PBKDF2 o scrypt), y un secreto guardado en el cliente.
    - Cifrado autenticado (AES-GCM, ChaCha20-Poly1305, etc.)
    - Uso de salt y nonce únicos cuando sean utilizados.

2. **Mensajería Cifrada sobre infraestructura existente** (e.j. Discord o Twitter)

- Implementar un sistema de envío/recepción de mensajes cifrados usando la API de una plataforma social.
- Alternativamente, implementar su propia infraestructura.
- Requisitos mínimos:
    - Generación de pares de llaves (RSA, ECC o Ed25519).
    - Aplicación que permita publicar llaves, elegir usuarios para enviar mensajes, leer mensajes recibidos, etc.
    - Cifrado híbrido (asimétrico para intercambio de llave, simétrico para el mensaje). Se puede usar alguno de los esquemas estudiados en clase (KEM/DEM).
    - Firmas digitales para autenticidad.
    - Codificación adecuada para enviar mensajes en texto plano (ej. Base64 o estilo PGP).
    - Validación de llave pública fuera de banda.

Entregables

La entrega del Proyecto estará dividida en dos. Cada entrega valdrá 10% del curso:

1. Entrega 1: Presentación de diseño (10-15 minutos), incluyendo diagrama de arquitectura, elección de algoritmos criptográficos, funcionalidad de aplicación, diseño de interfaz. La presentación será el Viernes de la semana 13.
2. Entrega 2: La entrega será el Viernes de la semana 16:
    - **Código fuente** en GitHub o repositorio similar.
    - **Informe técnico** que incluya:
        - Descripción del Sistema, incluyendo diagrama de arquitectura.
        - Justificación de algoritmos y librerías usadas.
        - Diagrama(s) de flujo criptográfico.
        - Análisis de amenazas y limitaciones.
    - **Presentación final** (10–15 minutos) donde el grupo explique el proyecto entero (resumen de diseño, modificaciones, etc. y detalles adicionales de implementación) y haga una demo.

Rúbrica de Evaluación 1ra entrega (20 puntos)

1. **Correctitud criptográfica (6 puntos)**
    - Uso adecuado de primitivos seguros, ya sean encriptaciones simétricas, hashes, o encriptaciones asimétricas.
    - Manejo correcto de llaves, nonces y sal.
    - Uso de modos de operación adecuados de ser aplicable.
2. **Diseño (6 puntos)**
    - Diseño de protocolos criptográficos adecuados.
    - Aplicación de defensas criptográficas donde sean necesarias.
    - Diagrama de arquitectura de aplicación.
    - Diseño de interfaz de aplicación.
    - Diagramas de flujo de data y procesos.
3. **Seguridad práctica (3 puntos)**
    - Consideración de amenazas reales.
    - Protección de datos en reposo y en tránsito.
4. **Presentación y comunicación (5 puntos)**
    - Claridad en la exposición oral.

- Claridad en material visual utilizado.
- Motivación de la implementación del proyecto.
- Explicación adecuada de todos los aspectos del diseño del proyecto.
- Explicación clara de decisiones criptográficas.
- Explicación de las funcionalidades de la aplicación, y los requisitos técnicos de implementación.

Rúbrica de Evaluación 2da entrega - tentativa (20 puntos)

1. **Correctitud criptográfica (4 puntos)**
    - Implementación adecuada de primitivos seguros, ya sean encriptaciones simétricas, hashes, o encriptaciones asimétricas.
    - Manejo correcto de llaves, nonces y sal.
    - Uso de modos de operación adecuados de ser aplicable.
2. **Diseño e implementación (5 puntos)**
    - Se incorporan las sugerencias de mejora de diseño (si las hubiera).
    - La implementación se adecúa al diseño.
    - Funcionalidad de la aplicación va de acuerdo con lo pactado en el diseño.
    - Se puede evidenciar la implementación de los protocolos criptográficos.
3. **Seguridad práctica (2 puntos)**
    - Implementación de medidas contra amenazas previamente consideradas.
    - Implementación de protección de datos en reposo y en tránsito.
    - Se hace uso de una herramienta de análisis estático de seguridad para analizar el código implementado.
4. **Presentación y comunicación (4 puntos)**
    - Claridad en la exposición oral.
    - Claridad en material visual utilizado.
    - Repaso de diseño.
    - Explicación de implementación.
    - Demo de aplicación, evidenciando funcionalidad y medidas criptográficas implementadas (e.j. data encriptada, contraseñas hasheadas, etc.)
5. **Informe técnico (5 puntos)**

    - Presentación escrita de motivación, diseño e implementación de la aplicación.
    - Claridad en las descripciones y explicaciones.
    - Claridad en diagramas utilizados.
    - Evaluación de resultados logrados, fallas y limitaciones.

Notas Importantes

- Se deben usar **librerías probadas** (ej. cryptography, PyNaCl, libsodium, etc.).
- Todos los algoritmos criptográficos deben ser elegidos de acuerdo con estándares actuales, e.j. no usar hashes rotos, o esquemas de encriptación débiles.
- Se puede implementar usando el lenguaje de programación y framework deseado.
- La aplicación puede ser móvil, de desktop, o Web.
- Es permisible el uso de IA para ayudar en toma de decisiones para diseño e implementación.
- Está prohibido implementar algoritmos criptográficos desde cero (ej. escribir tu propio AES).