ies zaidín-vergeles
granada

PumukyChat

Adrián Bertos Gómez

2ASIRB

Fernando Raya Díaz

I.E.S. Zaidín-Vergeles (GRANADA)

27/05/2025

Anexo - 3

Índice

1. Principios fundamentales del diseño

1.1. Separación entre contenido y claves

Una de las bases del diseño es desacoplar el mensaje cifrado de la clave que permite leerlo. Esta decisión:

- Permite almacenar un solo mensaje, independientemente del número de destinatarios.
- Hace posible revocar el acceso a un usuario simplemente eliminando su clave cifrada.
- Facilita escalar en grupos sin duplicar datos.

El mensaje se guarda una sola vez, y solo las claves AES se replican por cada usuario con acceso. Esta separación de responsabilidades mejora tanto el rendimiento como la trazabilidad del sistema.

1.2. Seguridad basada en el cliente

Todo el cifrado y descifrado ocurre exclusivamente en el navegador. El servidor no interviene ni puede acceder al contenido original. Esto garantiza que la seguridad no dependa del backend, ni de su configuración ni de su integridad.

Incluso si un atacante obtiene acceso a la base de datos o intercepta las comunicaciones, sin la clave privada del cliente no podrá descifrar nada útil.

2. Ventajas del enfoque híbrido

2.1. Escalabilidad en grupos

El cifrado híbrido permite reutilizar el mismo mensaje para todos los participantes, generando solo una clave AES por mensaje. La única parte que se personaliza por usuario es la clave cifrada.

Esto es especialmente útil en grupos, donde reenviar o cifrar múltiples veces el mismo mensaje sería ineficiente. Así se mantiene el rendimiento constante y se reduce el uso de recursos.

2.2. Aislamiento entre mensajes

Cada mensaje se cifra con una clave única e irrepetible. Aunque un atacante obtenga acceso a una clave AES concreta, solo podrá descifrar ese mensaje, no la conversación entera.

Esta fragmentación protege incluso frente a filtraciones parciales, y hace que comprometer la

privacidad de un usuario no implique exponer otros mensajes pasados ni futuros.

2.3. Aplicabilidad general

El mismo esquema sirve para cualquier tipo de contenido: texto, imágenes o archivos adjuntos. Esto evita mantener sistemas de cifrado distintos para cada caso, simplifica el desarrollo y asegura una protección uniforme en toda la plataforma.

3. Coste y rendimiento

3.1. Coste proporcional al número de receptores

El cifrado RSA solo se aplica sobre la clave AES (no sobre el mensaje completo), y solo una vez por destinatario. El coste escala de forma lineal con el número de usuarios, lo cual es perfectamente asumible para los casos reales del chat.

En grupos pequeños o medianos, el impacto es despreciable. En grupos grandes, podría optimizarse, pero no representa un cuello de botella crítico.

3.2. Sin dependencias externas

El sistema se basa únicamente en herramientas nativas del navegador: Web Crypto API e IndexedDB. No requiere librerías adicionales, extensiones, servidores intermedios ni frameworks externos.

Esto facilita la mantenibilidad del proyecto, reduce la superficie de ataque y evita problemas de compatibilidad entre plataformas.

4. Comparación con alternativas

4.1. Por qué no solo RSA

RSA es un algoritmo de clave pública pensado para datos pequeños. Cifrar mensajes directamente con RSA sería lento, produciría resultados grandes y no sería compatible con archivos o textos largos.

En cambio, al usar RSA solo para proteger la clave AES, se combinan las fortalezas de ambos algoritmos: seguridad asimétrica para distribuir claves y velocidad simétrica para cifrar datos.

4.2. Por qué no solo AES compartido

Si se usara una sola clave AES para todo un grupo o conversación, habría que confiar en el servidor para distribuirla, o compartirla previamente de forma segura entre los usuarios. En ambos casos se rompe el modelo descentralizado.

El enfoque actual evita este problema cifrando directamente la clave para cada usuario, sin confiar en el servidor como intermediario de seguridad.

4.3. Por qué no PGP

PGP introduce una capa de complejidad innecesaria para este caso. Requiere formato específico, compatibilidad con clientes externos y conceptos avanzados como firmas, certificados y cadenas de confianza.

El sistema actual está hecho a medida, funciona en tiempo real y se integra perfectamente con la interfaz del navegador. Es más sencillo de mantener, más flexible y no necesita adaptar el flujo de usuario a herramientas externas.

5. Limitaciones actuales

5.1. Metadatos no cifrados

El sistema protege el contenido, pero no oculta todos los metadatos. Información como:

- El nombre del grupo
- Quién ha enviado el mensaje
- Cuándo fue enviado
- Cuántos usuarios hay

se guarda sin cifrar. Esto permite mejorar el rendimiento y mantener funcionalidades como búsquedas o sincronización, pero también deja visibles ciertos patrones de uso.

5.2. Pérdida de la clave privada

Actualmente, la clave privada del usuario se guarda solo en su navegador. Si se borra (por limpieza de datos o pérdida del dispositivo), los mensajes anteriores no podrán recuperarse.

Esto es una garantía de seguridad, pero puede ser un problema si no se advierte al usuario. En futuras versiones se podría permitir una copia cifrada con contraseña, o una exportación manual.

5.3. Sin firma digital

Aunque AES-GCM asegura que los datos no se han modificado (autenticación implícita), no se utiliza una firma digital explícita con la clave RSA del remitente.

Esto significa que el receptor no puede verificar la autoría del mensaje de forma fuerte (no repudio). No es un problema grave para mensajería personal, pero sí sería relevante en entornos con requisitos legales o forenses.

6. Posibles mejoras futuras

- Añadir firmas digitales opcionales para autenticar el origen de los mensajes.
- Implementar cifrado de metadatos como nombres de grupo o alias.
- Permitir backups cifrados de la clave privada mediante contraseña.
- Añadir verificación de integridad con hashes y firmas.

PumukyDev

PumukyDev

https://pumukydev.com

adrian.bertosgomez@gmail.com

Adrián Bertos Gómez

PumukyDev

PumukyDev