Universidad Carlos III

Curso Criptografía 2025-26

Desarrollo de una aplicación que utiliza criptografía

Curso 2025-26

GRUPO: **82**

Alumnos:

Alejandro Ros Quesada - 100522331 - 100522331@alumnos.uc3m.es

Pablo Pino Castillo - 100522129 - 100522129@alumnos.uc3m.es

Repositorio de Github:

https://github.com/pabblopino/P1_Cripto.git

1. **Propósito de la aplicación y estructura interna:**

   El propósito de nuestra aplicación es desarrollar un sistema de votación electrónica segura para nuestros usuarios, que permite a los usuarios registrarse, autenticarse y dar un voto cifrado de forma confidencial. El objetivo principal es asegurar la aplicación de la criptografía garantizando así la confidencialidad, autenticación e integridad de la información de nuestros usuarios.

Con nuestra aplicación permitimos:

- Registrar nuevos usuarios de forma segura.
- Autenticar usuarios existentes mediante contraseña protegida con hash y salt
- Emitir y almacenar votos cifrados simétricamente
- Recuperar o actualizar votos seguramente con verificación criptográfica
- Registrar todos los eventos relevantes en un archivo log

Estructura Interna:

El proyecto está dividido en diferentes módulos:

- db.py: Que crea y gestiona la base de datos SQLite, incluyendo así las tablas de usuarios y votos.
- usuarios.py: Gestiona el registro y autenticación de los usuarios, haciendo hash seguro de sus contraseñas.
- votos.py: Se encarga del cifrado, descifrado y almacenamiento de los votos usando AES-GCM
- main.py: Módulo principal que controla el flujo del programa, las opciones del menú y las diferentes interacciones con el usuario

Todos los módulos utilizan logging para registrar información relevante que será de utilidad para los administradores sobre registros, fallos, horas… en datos/app.log

2. **Autenticación de usuarios:**

   Para la autenticación se realiza mediante una contraseña.

Algoritmos utilizados:

Usamos PBKDF2-HMAC-SHA256, un algoritmo para derivar hashes de contraseñas

Para su utilización hemos usado diversos parámetros:

- Iteraciones: 10000
- Salt aleatorio de 16 bytes generado con os.unrandom()
- Y una longitud del hash de 32 bytes

El hash resultante y el salt se almacenan en la base de datos junto al email del usuario

Gestión de contraseñas:

Las contraseñas por seguridad no se guarda directamente, por lo que en el registro, la contraseña se transforma mediante PBKDF2 y se guarda solo el hash y el salt, en el login, el sistema vuelve a calcular el hash y lo compara para validar el acceso.

Además de esto se obliga a hacer una validación de la contraseña que hace que deba tener como mínimo una letra mayúscula, una minúscula, un número y 8 caracteres de longitud.

Este mecanismo con SALT es realmente útil ya que evita los ataques de fuerza bruta o diccionario contra nuestra base de datos con métodos como tablas Rainbow, ya que si la misma contraseña se cifra

con un SALT distinto, dará lugar a otro hash distinto. Por ello es que las contraseñas de nuestra base de datos quedan protegidas, incluso si la base de datos se ve comprometida.

3. Cifrado simétrico/asimétrico

En nuestra aplicación usamos cifrado simétrico autenticado para proteger los votos de los usuarios.

Algoritmo utilizado:

Hemos usado AES-GCM con clave de 256 bits. Esta elección garantiza la confidencialidad e integridad al mismo tiempo, se considera uno de los modos más seguros y eficientes actualmente.

Generación y gestión de claves:

La clave simétrica: se genera AESGCM.generate_key(bit_length=256) al inicia la aplicación, almacenamos en un archivo "seguro" (más tarde en el trabajo haremos con claves públicas y privadas que esto sea mucho más seguro (cifrado asimétrico)) para utilizarla en futuras ejecuciones.

Cada voto se cifra al introducirlo y sólo puede descifrarse mediante la misma clave AES que se generó en la inicialización del sistema.

Con el cifrado asimétrico: Se plantea la inclusión de este en la segunda parte de la evaluación para mejorar la seguridad y facilitar la firma digital y el intercambio seguro de claves (claves públicas y privadas)

Prevemos que se empleará para proteger las claves AES de sesión, cifrándolas con la clave pública de un usuario y firmaremos digitalmente los votos, garantizando la autenticidad.

Se podría usar RSA con clave de 2048 bits o ECC para mayor eficiencia y menor tamaño de clave.

La gestión de claves: las claves asimétricas se generan por usuario, la clave privada se almacenará de forma segura, protegida por contraseña y la clave pública será disponible para cualquier persona.

Ventajas de usarlas combinadas: Con el cifrado simétrico tendremos una gran rapidez y eficiencia si así ciframos los datos. Con el cifrado asimétrico tendremos amyor seguridad en la gestión de claves. Y por último una buena escalabilidad y trazabilidad mediante los certificados digitales.

4. Autenticación de mensajes (MAC)

Con nuestra aplicación no contemplamos un HMAC separado, ya que el propio AES-GCM proporciona una autenticación integrada mediante un tag, lo que permite detectar cualquier alteración del texto cifrado y garantiza la integridad y autenticidad sin usar claves adicionales.

Ventajas del cifrados autenticado (AES-GCM):

Simplifica la gestión de claves, ya que la misma clave protege y autentica, proporciona integridad criptográfica a los fatos y es más eficiente que combinar AES con un HMAC separado.

En los logs registramos: el algoritmo utilizado (AES-GCM), longitud de la clave (256 bits) y AAD empleado en la operación

PRUEBAS

Para el desarrollo de pruebas hemos decidido testear nuestra aplicación probando varias de las funcionalidades:

Prueba de registro:

```
==================================
🗳️ SISTEMA DE VOTACIÓN SEGURA 🗳️
==================================

Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 1
Nombre: Alejandro
Email: 100522331@alumnos.uc3m.es
Contraseña: Cripto_mola123
Usuario registrado correctamente.
```

Prueba de voto:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 2
Email: 100522331@alumnos.uc3m.es
Contraseña: Cripto_mola123
Login del usuario correcto.
Introduce tu voto: hash_4life
Voto cifrado y almacenado correctamente en la base de datos.
   Cifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
Voto cifrado y registrado correctamente.
```

Ver voto cifrado:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 2
Email: 100522331@alumnos.uc3m.es
Contraseña: Cripto_mola123
Login del usuario correcto.
Ya tienes un voto registrado.
1. Ver voto
2. Cambiar voto
3. Cancelar
> 1
Voto descifrado correctamente.
   Descifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
Tu voto actual es: hash_4life
```

Cambio de voto:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 2
Email: 100522331@alumnos.uc3m.es
Contraseña: Cripto_mola123
Login del usuario correcto.
Ya tienes un voto registrado.
1. Ver voto
2. Cambiar voto
3. Cancelar
> 2
Introduce tu nuevo voto: Contraseñas_seguras
Voto cifrado y almacenado correctamente en la base de datos.
   Cifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
Voto actualizado correctamente.
```

Comprobar que ha cambiado el voto:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 2
Email: 100522331@alumnos.uc3m.es
Contraseña: Cripto_mola123
Login del usuario correcto.
Ya tienes un voto registrado.
1. Ver voto
2. Cambiar voto
3. Cancelar
> 1
Voto descifrado correctamente.
   Descifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
Tu voto actual es: Contraseñas_seguras
```

Cerrar sesión:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 3
Gracias por usar el Sistema de Votación Segura.
```

PRUEBAS ERRÓNEAS:

Usuario o contraseña no válida:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 2
Email: 10052231@alumnos.uc3m.es
Contraseña: Cripto_mola123
Email o contraseña incorrectos.
```

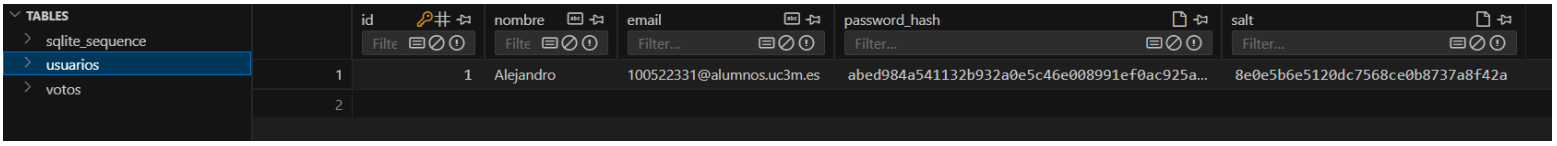
Volver a poner el mismo correo:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 1
Nombre: Alejandro
Email: 100522331@alumnos.uc3m.es
Contraseña: Crripto_mola123
Error: el email ya ha sido registrado con otro usuario.
```

Registro del log:

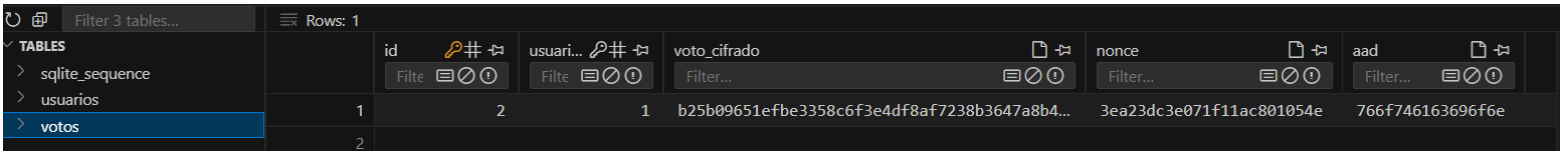
```
2025-10-29 20:18:37,904 - INFO - Clave AES generada y almacenada en c:\Users\USUARIO\Documents\Alejandro\UNI\Asignaturas\Criptografia\P1_Cripto\datos\clave_aes.key
2025-10-29 22:06:02,106 - INFO - Usuario registrado correctamente: 100522331@alumnos.uc3m.es
2025-10-29 22:06:27,067 - INFO - Inicio de sesi□n correcto: 100522331@alumnos.uc3m.es
2025-10-29 22:07:08,736 - INFO -  Voto cifrado con AES-GCM (256 bits) almacenado para usuario ID 1.
2025-10-29 22:08:09,337 - INFO - Inicio de sesi□n correcto: 100522331@alumnos.uc3m.es
2025-10-29 22:08:13,361 - INFO - Descifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
2025-10-29 22:08:38,228 - WARNING - Inicio de sesi□n fallido: 10052231@alumnos.uc3m.es
2025-10-29 22:09:18,537 - INFO - Inicio de sesi□n correcto: 100522331@alumnos.uc3m.es
2025-10-29 22:13:08,056 - INFO -  Voto cifrado con AES-GCM (256 bits) almacenado para usuario ID 1.
2025-10-29 22:13:35,603 - INFO - Inicio de sesi□n correcto: 100522331@alumnos.uc3m.es
2025-10-29 22:13:40,430 - INFO - Descifrado: Algoritmo AES-GCM | Longitud de clave: 256 bits | AAD: 'votacion'
2025-10-29 22:13:42,882 - INFO - Aplicaci□n finalizada por el usuario.
2025-10-29 22:56:24,686 - INFO - Clave AES generada y almacenada en c:\Users\USUARIO\Documents\Alejandro\UNI\Asignaturas\Criptografia\P1_Cripto\datos\clave_aes.key
2025-10-29 22:56:52,167 - WARNING - Intento de registro duplicado: 100522331@alumnos.uc3m.es
```

Base de datos:



podemos observar que aunque se filtre la contraseña no se puede conseguir gracias al hash

Votos:



Como vemos los votos están cifrados y no se pueden ver

Salir del programa:

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 3
Gracias por usar el Sistema de Votación Segura.
```

Interrupción del programa (Ctrl+C):

```
Menú principal:
1. Registrar usuario
2. Iniciar sesión y votar
3. Salir
> 1
Nombre:
 Programa interrumpido por el usuario.
```