Trabajo Práctico – Seguridad en Sistemas Operativos

Cifrado de Datos

Alumnos:

Keyla Valdes – valdeskeyla05@gmail.com

Andrés Vizone – andresvizone@gmail.com

Materia: Sistemas Operativos

Profesor: Osvaldo Falabella

Fecha de Entrega: 05 de junio de 2025

Índice

Introducción

El cifrado de datos es una técnica fundamental dentro de la seguridad informática, cuyo propósito es proteger la información de accesos no autorizados mediante la transformación de los datos en un formato ilegible. Esta práctica se ha vuelto cada vez más relevante en la era digital, donde la transmisión constante de información a través de redes y sistemas informáticos requiere mecanismos sólidos para garantizar la privacidad y la integridad de los datos. Sin el uso adecuado de técnicas de cifrado, la información podría quedar expuesta a ataques cibernéticos, comprometiendo su confidencialidad.

Este tema ha sido elegido porque tiene una gran importancia en la formación de un técnico en programación. Conocer cómo funciona el cifrado de datos permite a los programadores desarrollar sistemas más seguros y proteger información valiosa. Además, el aprendizaje de los algoritmos criptográficos ayuda a prevenir ataques cibernéticos y garantiza que los datos sensibles no sean comprometidos.

El objetivo de este trabajo es entender los conceptos básicos del cifrado de datos, explorando los diferentes tipos de cifrado, como el simétrico y el asimétrico, y sus aplicaciones prácticas. También se analizará el impacto que tiene la criptografía en la seguridad de los sistemas y se conocerán algunas herramientas utilizadas para su implementación. A través de este estudio, se busca fortalecer el conocimiento en seguridad informática y aprender estrategias para mantener la confidencialidad e integridad de la información.

Marco Teórico

Concepto de Cifrado y Criptografía

El cifrado es un proceso esencial en la seguridad informática que transforma información legible en un formato codificado para impedir accesos no autorizados. Se basa en algoritmos matemáticos que protegen la confidencialidad de los datos, asegurando que solo quienes posean la clave adecuada puedan interpretarlos.

Históricamente, el cifrado ha sido una herramienta clave para preservar la privacidad de la información. Desde métodos manuales como el cifrado César en la antigua Roma, hasta los modernos sistemas computacionales que protegen las comunicaciones digitales, el principio permanece: evitar que terceros accedan a datos sensibles sin autorización. En el ámbito digital, el cifrado es indispensable para proteger datos almacenados, transacciones financieras, autenticaciones de usuarios, correos electrónicos y todo tipo de comunicación en redes.

El cifrado no solo garantiza la privacidad, sino que también preserva la integridad de la información, evitando modificaciones maliciosas. En un mundo hiperconectado y con creciente intercambio de datos, el cifrado se convierte en un pilar de la seguridad informática, protegiendo información personal y corporativa frente a amenazas como el robo de datos, la suplantación de identidad y el espionaje digital. Su implementación es vital en áreas que van desde aplicaciones de mensajería hasta almacenamiento en la nube y navegación segura por internet.

Tipos de cifrado y sus características

Para entender los métodos de cifrado es necesario conocer sus dos principales formas:

Cifrado simétrico: También llamado cifrado de clave privada o de secretos compartidos, utiliza una única clave para cifrar y descifrar los datos. El receptor debe poseer la misma clave que el remitente para acceder al contenido. Su rapidez y eficiencia lo hacen ideal en muchos casos, aunque presenta el desafío de transferir la clave de forma segura sin que terceros la intercepten.

Cifrado asimétrico: Emplea un par de claves matemáticamente vinculadas: una clave pública, accesible a todos, y una clave privada, conocida solo por su propietario. El remitente cifra el mensaje con la clave pública del destinatario, quien lo descifra con su clave privada. Aunque es más lento que el simétrico, ofrece mayor seguridad en la transmisión y comunicación.

Tabla comparativa: Cifrado Simétrico vs Cifrado Asimétrico

Característica	Cifrado simétrico	Cifrado asimétrico
Numero de claves	Una única clave para cifrar y descifrar	Par de claves: pública y privada
Seguridad	Menor seguridad en transferencia de clave	Mayor seguridad en la transmisión
Velocidad	Muy rápido	Mas lento
Uso Común	Cifrado de grandes volúmenes de datos	Transmisión segura, firma digital
Ejemplos	DES, 3DES, AES	ECC, RSA
Dificultad de gestión	Fácil gestión, pero clave vulnerable	Más compleja gestión, pero clave pública libre

Algoritmos de cifrado más utilizados

AES (Advanced Encryption Standard)

AES es un algoritmo simétrico ampliamente utilizado, adoptado por el gobierno de Estados Unidos y muchas organizaciones. Presentado en 2001 por el NIST como reemplazo del DES, cifra datos en bloques de 128 bits con claves de 128, 192 o 256 bits.

Aplicaciones:

- Protección de transacciones en línea.
- Seguridad en discos duros y dispositivos móviles.

- Protección en aplicaciones de mensajería como WhatsApp y Signal.
- Seguridad en dispositivos IoT.

RSA (Rivest-Shamir-Adleman)

RSA es un algoritmo asimétrico creado en 1977, uno de los estándares más usados para la transferencia segura de datos. Usa clave pública para cifrar y clave privada para descifrar, y es matemáticamente seguro.

Aplicaciones:

- Firmas digitales.
- Seguridad en comunicaciones online.
- Cifrado en transacciones bancarias.
- Protección en sectores como salud y gobierno.

RSA también se utiliza para firmar digitalmente documentos, garantizando autenticidad y no repudio.

Otros algoritmos relevantes:

- DES (Data Encryption Standard): Algoritmo simétrico obsoleto, reemplazado por AES.
- 3DES (Triple DES): Versión mejorada de DES, aún usada en finanzas.
- Twofish: Algoritmo rápido y eficiente, popular en software libre.
- RC4: Usado en protocolos como WEP y WPA.

También existen los Criterios Comunes (CC), normas internacionales para evaluar la seguridad de productos tecnológicos.

Tabla: Algoritmos de cifrado más utilizados

Algoritmo	Tipo de cifrado	Tamaño de clave (bits)	Aplicaciones comunes	Ventajas	Desventajas
AES	Simétrico	128, 192, 256	Cifrado de discos, mensajería segura, IoT	Muy seguro, rápido, estándar	Requiere gestión segura de clave
RSA	Asimétrico	1024, 2048, 4096	Firmas digitales, transmisión segura	Alta seguridad, ampliamente utilizado	Más lento, consume recursos
DES	Simétrico	56	Sistemas antiguos (ya obsoleto)	Fácil de implementar	Vulnerable, no recomendado
3DES	Simétrico	112-168	Banca, aplicaciones legadas	Más seguro que DES	Más lento que AES
Twofish	Simétrico	Hasta 256	Software libre, cifrado de archivos	Rápido y seguro	Menos popular que AES

Herramientas de cifrado multiplataforma

Además de las soluciones nativas de los sistemas operativos, existen herramientas que funcionan en múltiples plataformas, facilitando la protección de datos en entornos heterogéneos.

GNU Privacy Guard (GPG)

Implementación libre del estándar OpenPGP para cifrado y firma digital.

Características:

- Cifrado y firma de correos.
- Soporta cifrado simétrico y asimétrico.
- Gestión avanzada de claves.
- Interfaz de línea de comandos.
  Ideal para usuarios con conocimientos técnicos que necesitan alta seguridad.
- Es compatible con Windows, macOS, Linux, RISC OS y Android

AxCrypt

Orientado a usuarios individuales y pequeños equipos.

Características:

- Cifrado de archivos individuales con AES-256.
- Integración con almacenamiento en la nube.
- Interfaz sencilla.
  Ideal para usuarios que buscan facilidad y rapidez en cifrado.
- Es compatible con Windows, macOS, Linux, iOS y Android

VeraCrypt

Herramienta de código abierto para cifrar volúmenes y particiones.

Características:

- Cifrado automático en tiempo real.
- Soporta AES, Serpent y Twofish.
- Permite volúmenes ocultos.
- Compatible con Windows, macOS y Linux.
  Ideal para cifrar grandes volúmenes de datos en empresas y usuarios avanzados.

Cryptomator

Diseñado para cifrar archivos en la nube.

Características:

- Compatible con Dropbox, Google Drive, OneDrive.
- Cifrado AES-256.
- Interfaz amigable.
- Multiplataforma (Windows, Linux, macOS, Android, iOS).
  Ideal para usuarios que almacenan datos en la nube y desean protección sencilla.

Cifrado en los principales sistemas operativos

1. Windows

Windows incluye múltiples tecnologías para proteger datos, dispositivos extraíbles, correos y archivos personales.

BitLocker

Según Microsoft BitLocker es una función de seguridad de Windows que ofrece cifrado para volúmenes completos, protegiendo contra amenazas como el robo de datos o la exposición derivada de dispositivos perdidos, robados o desechados de forma inadecuada. Es una herramienta integrada que cifra discos y unidades usando AES (modo XTS o CBC), con claves de 128 o 256 bits. Se complementa con TPM para validar la integridad del sistema en el arranque y permite autenticación multifactor. En ausencia de TPM, se usa contraseña o clave USB, aunque reduce seguridad.

Requiere partición separada sin cifrar y firmware compatible. En dispositivos nuevos, Windows configura automáticamente estas particiones.

BitLocker To Go

Extiende BitLocker a dispositivos extraíbles como USB o discos externos, usando contraseñas o certificados digitales.

Cifrado de dispositivo

Versión simplificada que cifra automáticamente la unidad del sistema y unidades fijas, activada en dispositivos que cumplen ciertos requisitos. Al configurar Windows por primera vez, se inicia el cifrado con clave almacenada en cuenta Microsoft o dominios empresariales.

Cifrado de datos personales

Protege archivos del usuario con autenticación moderna (PIN, biometría) usando Windows Hello. Las claves se almacenan de forma segura y se desbloquean al iniciar sesión.

Cifrado de correo electrónico

Windows soporta cifrado con S/MIME, Microsoft Purview y IRM para proteger mensajes y adjuntos, así como firmas digitales para autenticidad, disponibles en clientes como Outlook.

2. Linux

Linux ofrece un ecosistema potente y configurable para cifrar discos, particiones o archivos.

dm-crypt y LUKS

dm-crypt es un subsistema de cifrado integrado en el kernel Linux, compatible con varios algoritmos (AES, Serpent, Twofish).

LUKS añade gestión estandarizada para volúmenes cifrados, incluyendo almacenamiento de múltiples frases de contraseña y recuperación de claves. La herramienta cryptsetup facilita su uso.

Es común en distribuciones como Ubuntu y Debian para cifrar la partición raíz o directorios como /home.

EncFS

Sistema de cifrado en espacio de usuario para cifrar archivos individuales, montando un sistema de archivos virtual. Fácil de usar pero con menor rendimiento y seguridad cuestionada.

eCryptfs

Sistema de archivos criptográfico integrado en el kernel, cifrando directorios automáticamente

al acceder. Fue popular para cifrar directorios personales, aunque ha perdido soporte frente a soluciones más modernas.

3. macOS:

macOS protege los datos mediante FileVault, que cifra todo el volumen de arranque usando AES-XTS de 256 bits. FileVault requiere autenticación en cada inicio y desactiva el inicio de sesión automático para mayor seguridad.

En Macs con Apple Silicon o chip T2, el cifrado está integrado a nivel de hardware, utilizando claves almacenadas en el Secure Enclave o el chip T2, lo que aumenta la protección sin afectar el rendimiento.

El acceso al volumen cifrado está controlado por un **secure token** vinculado a la contraseña del usuario, que autoriza cambios críticos y desbloqueos.

Para recuperación, FileVault permite usar la cuenta de iCloud o una clave de recuperación personal, asegurando el acceso ante olvido de contraseña.

Además, FileVault bloquea accesos no autorizados al modo recuperación y permite borrar rápidamente los datos eliminando claves criptográficas, sin necesidad de sobrescribir físicamente el disco.

Sistema Operativo	Solución de cifrado principal	Tipo de cifrado	Características clave	Compatibilidad
Windows	BitLocker	Simétrico (AES 128/256)	Cifra discos completos, usa TPM, autenticación multifactor	Windows 10 Pro/Enterprise y superiores
Linux	dm-crypt + LUKS	Simétrico (AES, Serpent, Twofish)	Cifrado flexible para particiones, gestión avanzada de claves	Varias distribuciones (Ubuntu, Debian, Fedora, etc)
macOS	FileVault	Simétrico (XTS-AES 128)	Cifrado completo de disco, integración con T2 y Secure Enclave	macOS Sierra y versiones posteriores

Beneficios clave del cifrado

1. Seguridad de datos mejorada

El cifrado dificulta el acceso de hackers y usuarios no autorizados a información sensible. Es crucial para proteger datos confidenciales como tarjetas de crédito, registros médicos y números de seguro social, reduciendo el riesgo de robo de identidad.

2. Protección de información personal

En el mundo digital moderno, el cifrado protege datos almacenados en dispositivos móviles, servicios en la nube y cuentas en línea. Cuando se transmiten datos en Internet, como en

compras o banca digital, el cifrado los hace ininteligibles para los interceptores, garantizando la privacidad del usuario.

3. Privacidad mejorada en la comunicación

El cifrado de extremo a extremo asegura que solo el receptor legítimo pueda leer los mensajes, evitando el acceso de terceros. Esto es vital en conversaciones privadas, sobre todo en ámbitos financieros y legales. Aplicaciones como WhatsApp y Signal implementan este método para garantizar la seguridad en la mensajería.

4. Seguridad digital colectiva

Las prácticas de cifrado contribuyen a la seguridad digital en la protección de información almacenada y transmitida. Su integración dentro de los protocolos de seguridad fortalece la prevención contra ataques cibernéticos y pérdida de datos.

5. Transparencia y auditoría en el software

Los programas de cifrado pueden ser auditados por comunidades especializadas cuando son de código abierto. Esto permite verificar que el software funciona correctamente y no tiene vulnerabilidades ocultas. Sin embargo, el debate entre software propietario y código abierto sigue vigente en cuanto a control y accesibilidad de auditorías.

6. Resguardo contra software malicioso

El software privativo puede incluir funciones maliciosas que perjudican a los usuarios al restringir su control sobre sus propios datos. La Free Software Foundation ha documentado numerosos casos en los que el cifrado se vulnera intencionalmente. Un caso emblemático fue la solicitud a Apple para romper el cifrado de un usuario específico, demostrando la fragilidad del sistema ante presiones externas.

La implementación de cifrado en sistemas operativos garantiza que los datos sean accesibles únicamente por usuarios autorizados y reduce riesgos de violaciones de seguridad.

3. Caso Práctico

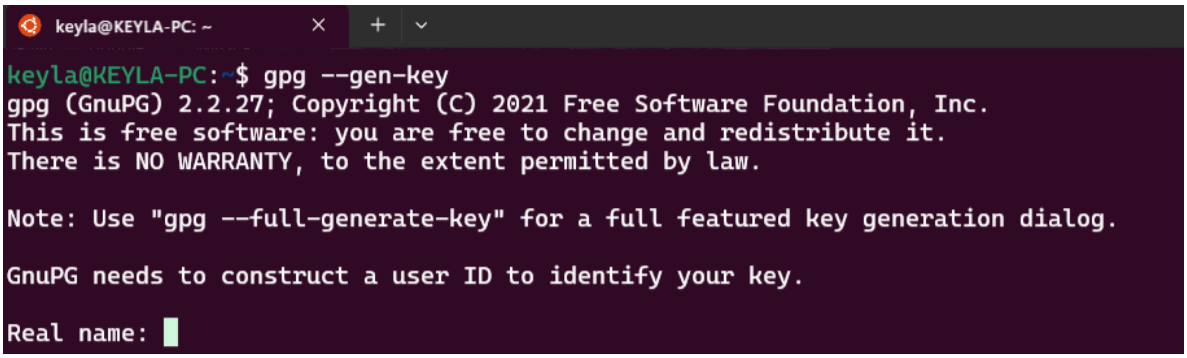
Descripción del problema:

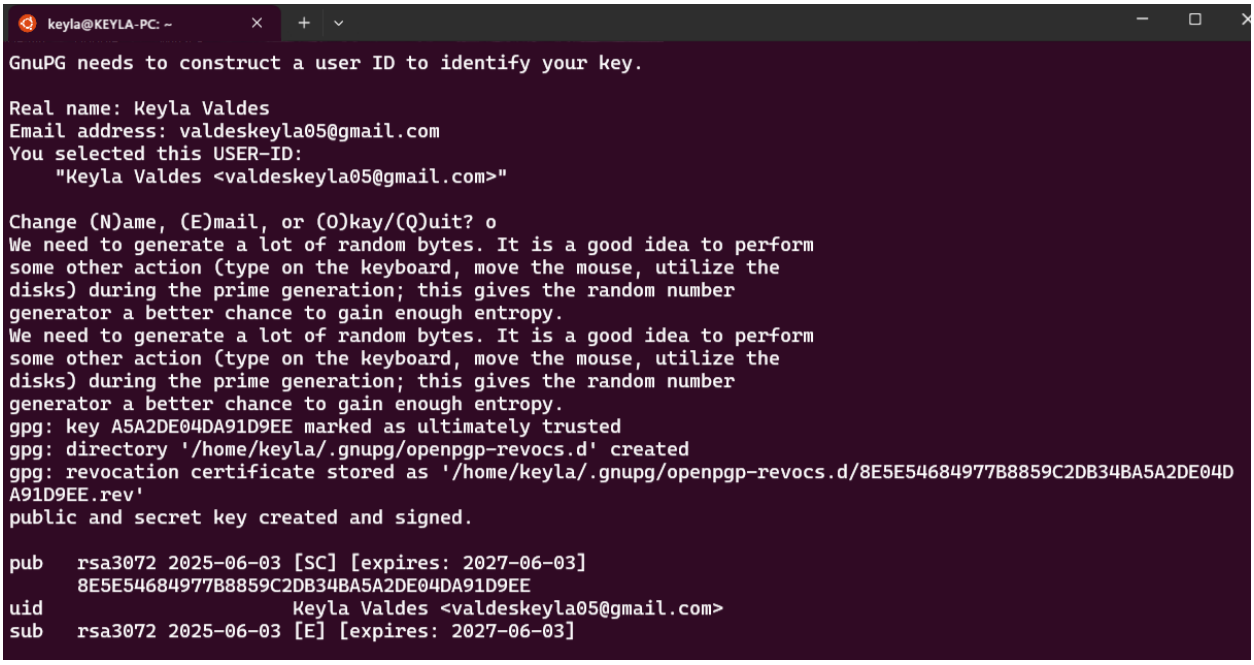
En este caso práctico se aborda la necesidad de proteger un archivo de texto que contiene información confidencial mediante técnicas de cifrado asimétrico y verificación de integridad. El objetivo es garantizar que solo el destinatario autorizado pueda acceder al contenido y que no se haya alterado durante el proceso.

Pasos realizados:

1. **Generación de claves GPG:**

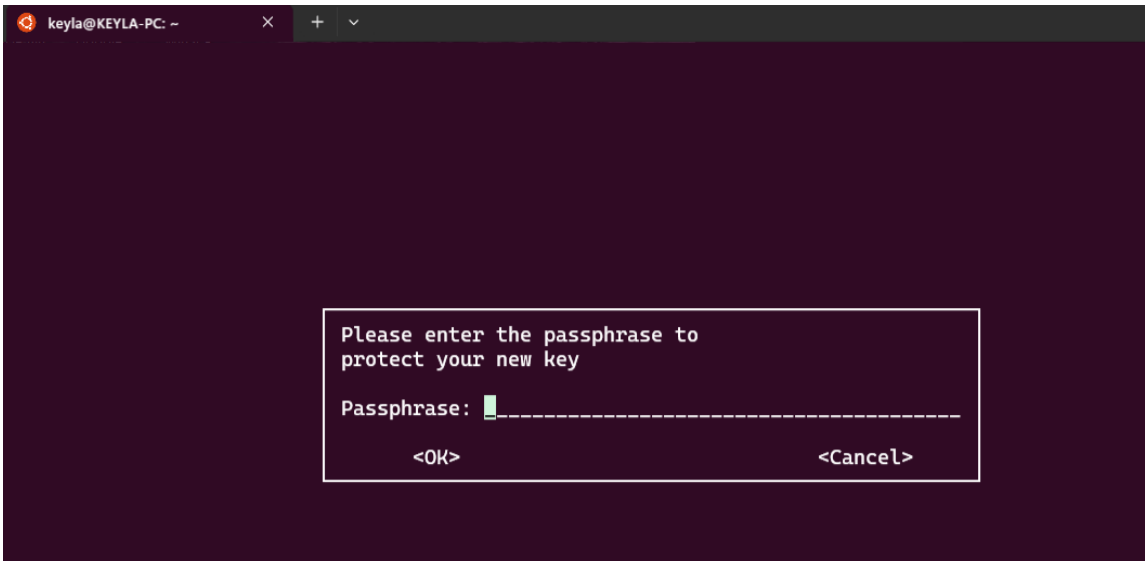
Se creó un par de claves (pública y privada) utilizando el comando gpg --gen-key, necesarias para aplicar criptografía asimétrica.





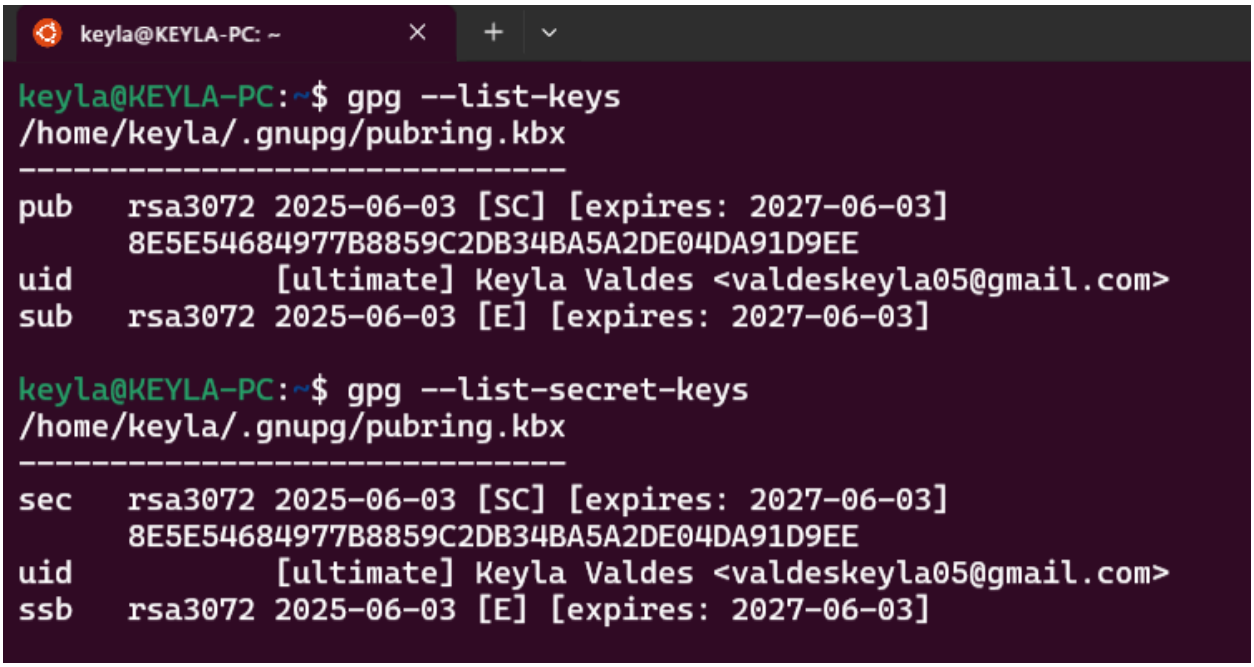
2. Creación de la frase de paso para la clave GPG

El sistema solicita una **frase de paso** para proteger la clave privada. Se introduce una contraseña segura que servirá como medida adicional de protección



3. Verificación de claves generadas:

Se usaron los siguientes comandos para comprobar que las claves se generaron correctamente:

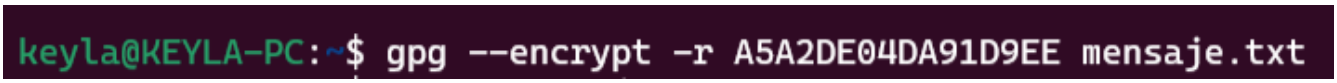


4. Creación del archivo a proteger:

Se creó un archivo llamado mensaje.txt con un mensaje confidencial mediante el comando:

5. **Cifrado del archivo:**

El archivo fue cifrado con la clave pública del destinatario:

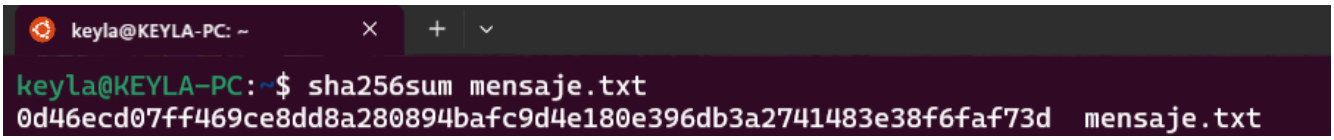


6. Verificación del archivo cifrado:

Se comprobó la existencia del archivo cifrado mensaje.txt.gpg y se verificó que su contenido no era legible, como resultado del cifrado.

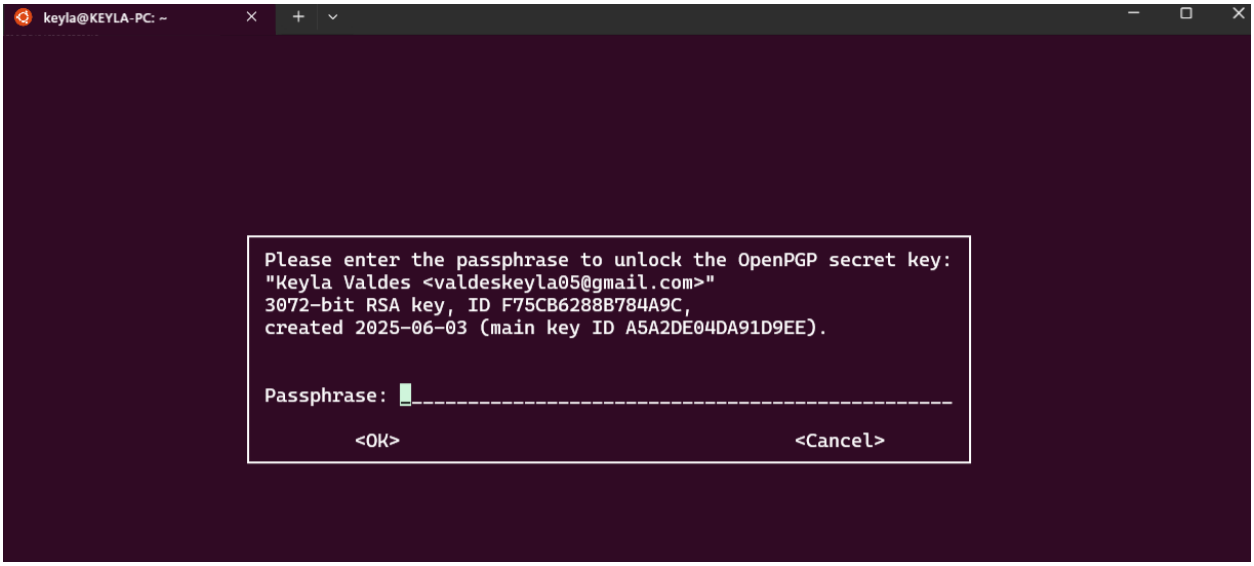
7. Generación de hash:

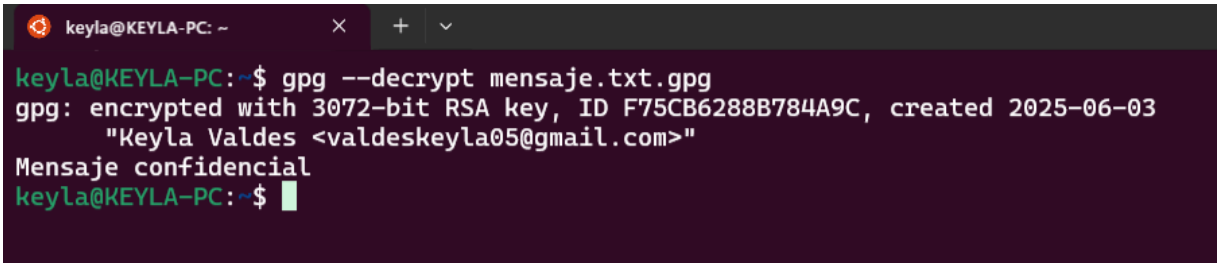
Se genera una suma de verificación (hash) del archivo original:



8. Descifrado del archivo:

Se descifra el archivo cifrado y se muestra el contenido directamente en la terminal, sin guardarlo en un archivo.

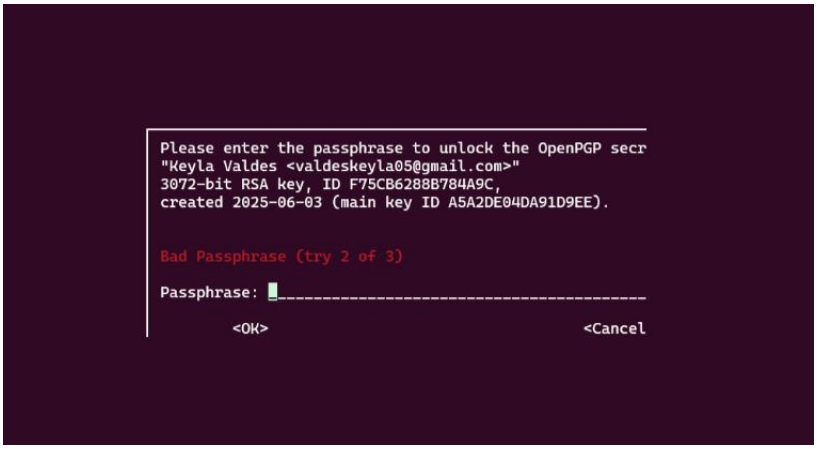


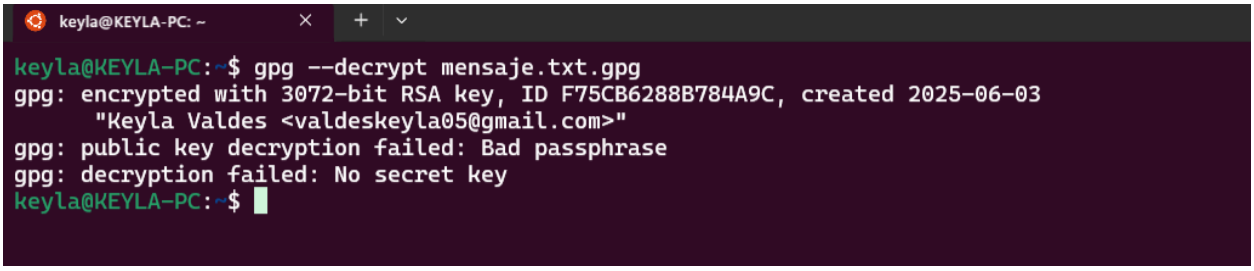


9. Prueba de error: uso de clave incorrecta para descifrar

Descripción:

Se intentó descifrar el archivo cifrado utilizando una clave privada que no correspondía a la clave pública con la que se había cifrado originalmente el archivo.





10. Descifrar el archivo y guardar el resultado en otro archivo

En este caso se descifra el archivo cifrado y se guarda el contenido resultante en un nuevo archivo.

11. Verificación de integridad:

Se comparó el hash del archivo descifrado con el original para confirmar que no hubo alteraciones:

Validación del funcionamiento:

El archivo fue exitosamente cifrado y descifrado. La integridad se verificó correctamente gracias a los hashes coincidentes antes y después del proceso.

Metodología utilizada

Investigación previa:

Para llevar a cabo el trabajo se consultaron diversas fuentes prácticas y tutoriales disponibles en línea, incluyendo blogs técnicos, foros de usuarios y videos explicativos. Se investigaron los conceptos clave de criptografía asimétrica, el uso de GPG en sistemas Linux, y herramientas de hashing como sha256sum. Las fuentes más útiles incluyeron tutoriales en YouTube y artículos de la comunidad.

Etapas del desarrollo:

- **Diseño:** Se definió un flujo simple: generar claves → cifrar archivo → verificar → descifrar → comprobar integridad.
- **Pruebas:** Se realizaron pruebas con diferentes algoritmos de hash y con claves distintas para comprobar el funcionamiento correcto.
- **Herramientas utilizadas:**
    - Sistema operativo: Linux (Ubuntu)
    - Terminal Bash

- GNU Privacy Guard (GPG)
- Herramientas de hashing integradas en Linux

Trabajo colaborativo:

El trabajo fue realizado por dos integrantes. Ambos colaboraron en la investigación y en la redacción del marco teórico. Si bien las capturas de pantalla fueron realizadas por una de las personas del grupo, el otro integrante participó activamente en el desarrollo del caso práctico y en la revisión de todo el trabajo para asegurar que estuviera completo y correcto.

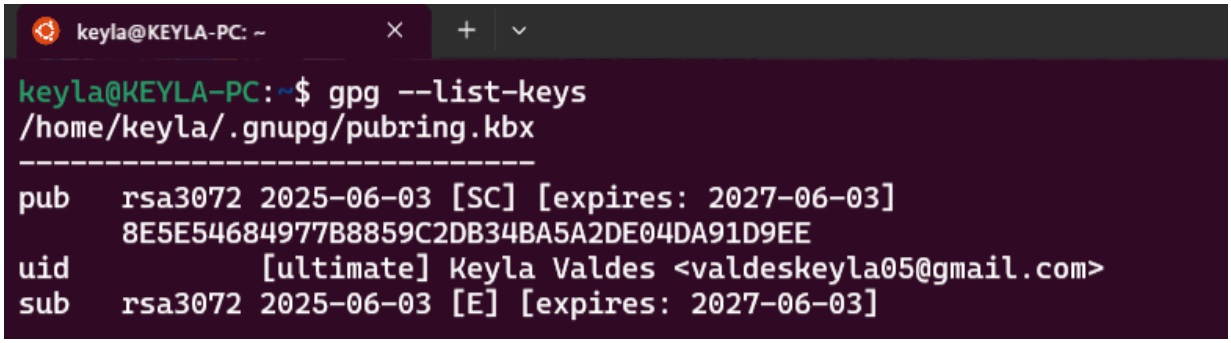
5. Resultados Obtenidos

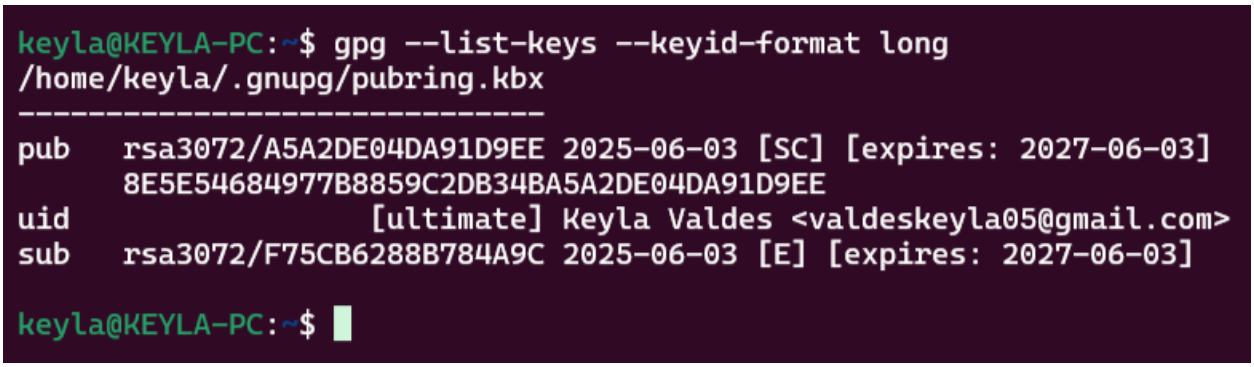
Logros:

- Se logró cifrar y descifrar correctamente el archivo utilizando claves generadas con GPG.
- Se realizaron pruebas adicionales, como intentar descifrar el archivo con una clave incorrecta, lo cual falló como se esperaba, demostrando la seguridad del cifrado asimétrico.
- Las herramientas de hashing (sha256sum) permitieron verificar la integridad del archivo antes y después del cifrado.
- El contenido original se mantuvo intacto después del proceso, comprobando que no se produjeron alteraciones.

Errores y dificultades:

- Al principio hubo dudas sobre cómo identificar correctamente el ID de la clave pública a utilizar con --encrypt. Inicialmente no se mostraba el formato completo del ID, lo que generó confusión. Esto se solucionó utilizando gpg --list-keys --keyid-format long, lo que permitió visualizar el identificador completo y usarlo correctamente en el cifrado.





Casos de prueba: Se realizaron múltiples pruebas con archivos distintos y configuraciones variadas de claves. En todos los casos, el proceso de cifrado y descifrado funcionó correctamente. También se compararon diferentes algoritmos de hash para observar sus salidas y tiempos de ejecución.

Comparación de las salidas de los hashes del mismo archivo (archivo.txt):

6. Conclusiones

Aprendizajes:

Se comprendió de forma práctica cómo funciona la criptografía asimétrica en Linux utilizando GPG.

Se aprendió a generar y manejar claves públicas y privadas, y a reconocer su rol en los procesos de cifrado y descifrado.

Se incorporó el uso de funciones de hashing para verificar la integridad de los datos, entendiendo las diferencias entre algoritmos como SHA256, MD5 y SHA512.

Dificultades

- La principal dificultad fue comprender el funcionamiento general de la criptografía asimétrica y cómo aplicar sus conceptos en la práctica. Esta dificultad se superó mediante la consulta de distintos tutoriales y la realización de pruebas en el entorno Linux.
- Al principio, también hubo confusión sobre las formas de descifrar archivos, ya que se pensaba que el comando mostraba y guardaba el contenido automáticamente, hasta que se aprendió a usar la redirección de salida para guardar el archivo descifrado.

Conclusión general

Este trabajo permitió aplicar de forma práctica conceptos fundamentales de la criptografía asimétrica mediante el uso de GPG en Linux. A lo largo de las distintas etapas, se aprendió a generar claves, cifrar y descifrar archivos, y verificar su integridad utilizando funciones de hashing. Además, se identificaron y resolvieron dudas comunes relacionadas con el manejo de claves y comandos. La experiencia fue útil para comprender cómo proteger información de manera segura y sentó una base sólida para futuras implementaciones más avanzadas o automatizadas.

Bibliografia

Protecting Data with FileVault — Deployment and Management Tutorials | Documentation

https://alexhost.com/es/faq/9-mejores-programas-de-cifrado-para-2025/
libro de seguridad de Windows 11: cifrado y protección de datos | Microsoft Learn

https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/

Comparing Encryption Tools: Analysis of Different Encryption Tools Available for Linux - JumpCloud

https://www.veritas.com/es/mx/information-center/encryption

https://latam.kaspersky.com/resource-center/definitions/encryption

Bibliografía (formato APA 7.ª edición)

- AlexHost. (2025). *9 mejores programas de cifrado para 2025*. Recuperado el 2 de junio de 2025, de https://alexhost.com/es/faq/9-mejores-programas-de-cifrado-para-2025/
- Artículo 19. (s.f). *Guía práctica: Seguridad en sistemas operativos – Cifrado*. Recuperado el 2 de junio de 2025 de https://seguridadintegral.articulo19.org/wp-content/uploads/2020/06/art19_2020_Guia_CifradoSO_V2.pdf
- JumpCloud. (2024). *Comparing Encryption Tools: Analysis of Different Encryption Tools Available for Linux*. Recuperado el 2 de junio de 2025, de https://jumpcloud.com/blog/comparing-encryption-tools-linux
- Kaspersky. (s.f.). *¿Qué es el cifrado y cómo funciona?*. Recuperado el 2 de junio de 2025, de https://latam.kaspersky.com/resource-center/definitions/encryption
- Microsoft. (2025). *Cifrado y protección de datos en Windows 11*. Microsoft Learn. Recuperado el 2 de junio de 2025, de https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/
- Veritas. (s.f.). *El arte del secreto: Entender las técnicas de cifrado*. Recuperado el 2 de junio de 2025, de https://www.veritas.com/es/mx/information-center/encryption

- Apple Inc. (s.f.). *Protecting Data with FileVault — Deployment and Management Tutorials*. Recuperado el 2 de junio de 2025, de https://it-training.apple.com/tutorials/deployment/dm210/