Investigación de Herramientas para Cifrado de Archivos y Carpetas

Autor – David Ricardo Ordoñez Mora

Objetivo de la Investigación

Analizar y justificar el uso de herramientas de cifrado de archivos para carpetas locales, enfocadas en criptografía simétrica realizable en Python, con énfasis en su implementación práctica.

Herramientas Investigadas

1. **Fernet** (biblioteca **cryptography**) - Fernet es un sistema de cifrado simétrico que utiliza AES-128-CBC + HMAC-SHA-256, con claves de 32 bytes, IV aleatorio y verificación de integridad.
    a. **Ventajas**:
        i. - Asegura confidencialidad, integridad y autenticidad con una API sencilla.
        ii. - Ideal para cifrar bytes de archivos completos.
    b. **Justificación**: Combina robustez (AES+CBC+HMAC) con facilidad de uso en Python, apropiado para proteger información local en contextos académicos y profesionales.
2. **PBKDF2** (derivación de clave por contraseña) - **PBKDF2** es una función estándar (RFC 2898, PKCS #5 v2.0) que transforma contraseña + sal + múltiples iteraciones en una clave simétrica segura.
    a. **Ventajas**:
        i. - Añade una capa de seguridad al convertir contraseñas en claves robustas.
        ii. - Ideal para usar con Fernet en escenarios donde la clave deriva de una contraseña de usuario.
    b. **Justificación**: Fortalece sistemas basados en password, evitando ataques de fuerza bruta.
    c. Video: https://www.youtube.com/watch?v=fKw4L-mmv0c PBKDF2 tutorial in Python – Python Basics
3. **Tkinter** - GUI estándar de Python, multiplataforma, ideal para prototipos rápidos.
    a. **Usos en proyecto:**
        i. - Ventana principal
        ii. Selección de archivos (filedialog)
        iii. Entradas de clave y botones de acción
        iv. Mensajes emergentes con messagebox
    b. **Justificación**: No requiere dependencia externa, permite construir interfaces funcionales sin complicaciones.
    c. **Video**: https://www.youtube.com/watch?v=ouO_QRJtvNE - Open Files Dialog Box – Python Tkinter GUI Tutorial #15
4. **PyInstaller** Genera ejecutables (e.g. .exe) de scripts Python.
    a. **Ventajas**:

i. Facilita distribución sin exigir Python en el sistema.
ii. Compatible con GUIs como Tkinter.

b. **Justificación:** Ideal para entregar aplicaciones a usuarios finales sin entornos de desarrollo.

5. **logging (Python estándar) -** Biblioteca para registrar eventos: errores, advertencias, información.
    a. **Implementación:** Errores durante cifrado/descifrado se guardan en logs/error.log
    b. **Justificación:** Mejora trazabilidad de errores y soporte técnico, manteniendo calidad y profesionalismo en el sistema.

Comparativa de herramientas

Herramienta	Cifrado seguro	Interfaz	Portable	Requiere instalación	Programable
Python + Fernet	✅ AES-128/CBC	✅ GUI	✅ .exe	❌ Python o exe	✅
VeraCrypt	✅ Muy fuerte	✅ GUI	❌	✅ Instalación	❌
7-Zip + AES	✅ Medio	❌ CLI	✅	✅ Instalación	❌
WinRAR con clave	❌ Menos fuerte	✅ GUI	✅	✅ Instalación	❌

Conclusión

La solución combinada:

1. **Fernet + PBKDF2:** cifrado fuerte y seguridad de clave
2. **Tkinter:** interfaz accesible sin dependencias externas
3. **PyInstaller:** ejecutable para usuarios finales
4. **logging:** calidad profesional y seguimiento de errores es una plataforma robusta, portable y adecuada para uso académico, personal y empresarial.