ASCII Signature Encryptor/Decryptor Python project using Tkinter and cryptography:

📌 Project Overview

This GUI tool allows users to:

1. Generate an **ASCII art signature** from their name and the current timestamp.
2. **Encrypt** this ASCII art using a short password and **save** it securely.
3. Later, the user can **decrypt** the file using the same password and view the original ASCII signature.

🔧 Modules & Libraries Used

Library	Purpose
tkinter	GUI creation (buttons, labels, text fields)
pyfiglet	Create stylized ASCII art from text
datetime	Add timestamp to the signature
cryptography	Perform secure encryption/decryption using Fernet & PBKDF2HMAC
os	Create directories, work with file paths
base64	Encode encryption keys safely for Fernet

🔐 Cryptographic Flow

password_to_key(password, salt)

- Uses PBKDF2HMAC to convert a **user password** into a 32-byte key.
- Key is safe to use with Fernet for encryption.
- Salt adds randomness and is stored in the encrypted file.

encrypt_text(text, key)

- Uses Fernet to **encrypt the ASCII signature** using the derived key.

decrypt_file_gui(file_path, password)

- Reads the salt + encrypted content from the file.

- Reconstructs the key from the password + salt.
- Decrypts the ASCII signature.

🖼 GUI Components

🔹 Top Fields

- name_entry: Input field to type the name.
- password_entry: Field for password input during encryption (hidden with *).

🔹 Buttons

- **Encrypt & Save Signature**:
  - Generates ASCII.
  - Encrypts using password.
  - Saves file and displays ASCII and encrypted string.
- **Decrypt Signature**:
  - Asks for file to decrypt.
  - Prompts password in popup.
  - Decrypts and displays ASCII.

🔹 Text Boxes

- ascii_text_box: Shows the generated ASCII art.
- encrypted_text_box: Shows base64-encrypted data.
- decrypted_text_box: Shows the decrypted signature after correct password.

📜 Code Section Breakdown

🔹 Import Libraries

import tkinter as tk

from tkinter import ...

- All GUI and crypto-related libraries.

🔹 **Crypto Functions**

def password_to_key(password, salt): ...

def encrypt_text(text, key): ...

def decrypt_file_gui(file_path, password): ...

- These handle key derivation, encryption, decryption.

🔹 **Signature Generator**

def generate_ascii_signature(name): ...

- Uses pyfiglet and adds a timestamp to make each signature unique.

🔹 **Encrypt Action**

def encrypt_action(): ...

- Gets name & password → creates ASCII → encrypts → saves to file → updates GUI boxes.

🔹 **Decrypt Action**

def decrypt_action(): ...

- File dialog opens → password is prompted → file decrypted and displayed.

🔹 **GUI Setup**

root = tk.Tk()

...

root.mainloop()

- Places all UI elements and controls window behavior.

✅ Features Summary

Feature	Description
Password Encryption	User provides password; key is derived securely
Salt-Based Protection	Random salt ensures security even with same password
Time-stamped ASCII Signatures	Makes each signature unique
File Dialog + Password Prompt	User-friendly decryption experience
Fixed Output Folder	Files saved automatically to a predefined directory
GUI Feedback	Success and error popups, auto-filled output boxes

---

💡 Suggestions for Improvement

Idea	Benefit
Add “Open Folder” after save	Quick access to encrypted file
Checkbox to show/hide password	Better user experience
Save encrypted+decrypted logs	Record history of actions
Export ASCII to .png (image)	Visual saving of signatures
