

Project Secret: Steganographic-Based Encryption System Documentation

August 29, 2025

1 Introduction

Project Secret is a Python-based steganographic encryption system designed to conceal sensitive information within text through a combination of term substitution and alphabet-based encryption. Integrated with a Flask web application, Secret enables users to encrypt, decrypt, and manipulate text by substituting sensitive terms (e.g., names, weapons, and violent terms) and applying a substitution cipher. This system is ideal for discreet communication, encoding messages to appear innocuous while preserving their meaning for authorized recipients.

This document outlines the functionality, technical implementation, usage instructions, and testing procedures for Project Secret, including test data for substitution lists. It targets developers, users, and administrators interacting with or maintaining the system.

2 System Overview

Project Secret employs two layers of concealment:

- **Steganographic Substitution:** Replaces sensitive terms with harmless ones, such as boy's names (e.g., "Ahmed") with girl's names (e.g., "Ayesha"), weapons (e.g., "Pistol") with flowers (e.g., "Rose"), and violent terms (e.g., "kill") with benign actions (e.g., "cooking"). This disguises the text's true meaning.
- **Substitution Cipher:** Applies an alphabet-based encryption (e.g., 'a' → 'Z', 'b' → 'Y') to further obscure the text, ensuring unreadability without the decryption key.

The system comprises two main components:

- **substitution_cipher.py:** Implements the core logic, including the NameChanger class for substitutions and the SubstitutionCipher class for encryption and decryption.
- **app.py:** A Flask web application providing a user interface via JSON requests to perform Secret's operations.

Secret supports four operations:

1. **Encrypt:** Substitutes sensitive terms and applies alphabet encryption.
2. **Simple Decrypt:** Reverses alphabet encryption, preserving substituted terms (e.g., “Ayesha” for an encrypted “Ahmed”).
3. **Full Decrypt:** Reverses both encryption and substitutions, restoring the original text (e.g., “Ahmed”).
4. **Reverse Substitutions:** Swaps substituted terms to originals without affecting encryption (e.g., “Ayesha” to “Ahmed”).

3 Technical Implementation

3.1 Substitution Logic

The NameChanger class in Project Secret maintains substitution lists for names, weapons, and violent terms. The `substitute_names_in_text` method splits text into words, applying substitutions based on the operation (encryption or decryption). It preserves case and punctuation for natural text output.

3.2 Encryption and Decryption

The SubstitutionCipher class uses a fixed key (“ZYXWVUTSRQPONMLKJIHGFEDCBA”) to map letters (e.g., ‘a’ → ‘Z’). The `encrypt` method substitutes terms then encrypts the text. The `decrypt` method reverses alphabet encryption, producing substituted text. The `full_decrypt` method further reverses substitutions to restore the original text.

3.3 Flask Integration

The Flask application (`app.py`) provides:

- GET `/`: Serves `index.html` for user input.
- POST `/process`: Processes JSON requests with `text` and `action` fields, supporting Secret’s operations: “`encrypt`”, “`simple_decrypt`”, “`full_decrypt`”, and “`reverse_sub`”.

Logging at the DEBUG level tracks operations and errors.

4 Usage Instructions

4.1 System Requirements

- Python 3.8+
- Flask (`pip install flask`)
- Web browser

4.2 Setup

1. Save `substitution_cipher.py` and `app.py`.
2. Create a `templates` folder with `index.html`.
3. Install Flask: `pip install flask`.
4. Run: `python app.py`.

4.3 Web Interface Usage

1. Navigate to `http://localhost:5000`.
2. Enter text (e.g., “Ahmed has a Pistol.”).
3. Select an action (Encrypt, Simple Decrypt, Full Decrypt, Reverse Substitutions).
4. Submit to view the result (e.g., “Zvhbz szh z Ilhv.” for Encrypt).

4.4 Command-Line Usage

1. Run `python substitution_cipher.py`.
2. Select an option (1: Encrypt, 2: Full Decrypt, 3: Simple Decrypt, 4: Reverse Substitutions, 5: Exit).
3. Enter text and view the output.

4.5 Example

- **Input:** “Ahmed has a Pistol.”
- **Encrypt:** “Zvhbz szh z Ilhv.” (after “Ahmed” → “Ayesha”, “Pistol” → “Rose”)
- **Simple Decrypt:** “Ayesha has a Rose.”
- **Full Decrypt:** “Ahmed has a Pistol.”
- **Reverse Substitutions:** “Ahmed has a Pistol.”

5 Testing and Validation

5.1 Test Cases

Project Secret’s `test_cipher` function validates functionality with cases like:

- Encrypt “Ahmed has a Pistol.” → “Zvhbz szh z Ilhv.”
- Simple Decrypt “Ayesha has a Rose.” → “Ayesha has a Rose.”
- Full Decrypt “Ayesha has a Rose.” → “Ahmed has a Pistol.”

- Reverse Substitutions “Ayesha has a Rose.” → “Ahmed has a Pistol.”

Run tests via option 5 in the command-line menu. Logs detail substitution and encryption steps.

5.2 Test Data

The substitution lists serve as test data to verify Secret’s behavior. Below are the key substitution pairs:

Table 1: Name Substitutions for Project Secret

Original (Boy’s Name)	Substituted (Girl’s Name)
Ahmed	Ayesha
Ahmad	Aisha
Muhammad	Fatima
Ali	Mariam
Hassan	Zainab
Usman	Hira
Bilal	Aiza
Omar	Sana
Ibrahim	Amna
Hamza	Sara
Zain	Mahnoor
Rayyan	Alishba
Haris	Zunaira
Saad	Laiba
Farhan	Anaya
Taha	Hania
Yusuf	Dua
Abdullah	Mehak
Arham	Rida
Zayan	Zoya
Shayan	Kiran
Uzair	Tazeen
Tariq	Minahil
Mahinn	Saima
Qaiser	Zareen
Essa	Tehzeeb
Musa	Saman
Dawood	Ammarah
Shoiab	Fatima

Table 2: Weapon and Violent Term Substitutions
for Project Secret

Original Term	Substituted Term
Pistol	Rose
Glock	Tulip
Desert Eagle	Lily
Revolver	Wisteria
Colt Python	Yarrow
Rifle	Aster
AK-47	Azalea
M16	Bird of Paradise
Sniper rifle	Black-Eyed Susan
Barrett .50 cal	Bleeding Heart
Shotgun	Bougainvillea
Remington 870	Calla Lily
Benelli M4	Canterbury Bells
Submachine gun	Cherry Blossom
MP5	Columbine
Uzi	Cornflower
Machine gun	Cowslip
M249 SAW	Cyclamen
M60	Dandelion
Grenade launcher	Edelweiss
M203	Forget-Me-Not
RPG-7	Freesia
Cannon	Fuchsia
Mortar	Gaillardia
Howitzer	Geranium
Anti-tank missile	Goldenrod
Javelin	Hellebore
knife	white
gun	blue
tank	pinkrose
dagger	blackrose
bloody	walking
blood	jumping
kill	cooking
killing	reading
killed	writing
shot	painting
shooting	swimming
stabbed	climbing
stab	skipping
stabbing	dancing
hunt	running
hurt	singing

Table 2: Weapon and Violent Term Substitutions
for Project Secret

Original Term	Substituted Term
betrayed	sleeping
hate	laughing
punch	playing
beat	working
beating	traveling
hit	studying
hitting	eating
assault	eating
assaulted	dancing
sexual	running
violence	singing
violent	sleeping

6 Conclusion

Project Secret provides a robust solution for concealing sensitive information through steganographic substitution and alphabet encryption. The Flask interface ensures accessibility, while the command-line interface supports testing. The test data validates reliability across various inputs. Future enhancements could include dynamic keys or expanded substitution categories to further strengthen Secret’s capabilities.