DAV UNIVERSITY

Empowering Students with 21st century Skills

STEG-X

Submitted in the partial fulfilment of the requirement for the award of degree of
Bachelor of Technology
in
Computer Science and Engineering
Batch
(2023-2027)

Submitted to:

Ms. Ananya Sharma

Submitted by:

Kunal Manhas

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DAV UNIVERSITY
JALANDHAR- PUNJAB 144012

STEG-X: An Advanced Steganography Tool

45 days Industrial Training Project – Coder Roots

By: Kunal Manhas

Reg no: 12300053

Github link: https://github.com/Kunal-Manhas/STEG-X-

Unveiling the Invisible: Understanding Steganography

Steganography is an ancient art reimagined for the digital age – the practice of concealing secret data within ordinary, innocuous files, making its very existence undetectable to unintended observers. Unlike cryptography, which scrambles messages, steganography ensures the message itself remains hidden in plain sight.

STEG-X harnesses this principle, specifically employing the Least Significant Bit (LSB) technique. This method subtly alters the lowest-order bits of pixel data in an image, embedding information without visibly changing the image itself. It's a powerful tool for covert communication in a world of pervasive surveillance.

STEG-X: Your Covert Communication Toolkit

STEG-X is a robust, Python-based cybersecurity tool developed over 45 intensive days at Coder Roots. It provides a comprehensive solution for digital information hiding, extraction, and detection. Our primary objective was to create a versatile application capable of:

Hiding Messages

Discreetly embed text within digital images.

Extracting Secrets

Retrieve hidden messages from steganographic images.

Detecting Anomalies

Analyze images for subtle LSB manipulations, exposing hidden content.

This project bridges theoretical cybersecurity concepts with practical application, offering both a command-line interface for technical users and a graphical user interface for broader accessibility.

Project Objectives: Crafting a Secure & User-Friendly Tool

Our development journey for STEG-X was guided by clear, actionable objectives focused on delivering a high-quality, functional cybersecurity utility. Each goal contributed to the tool's overall effectiveness and user experience.

01

Python-Based Cybersecurity Tool

Develop a reliable and efficient cybersecurity application entirely in Python.

02

Secure LSB Implementation

Engineer robust message hiding capabilities using the Least Significant Bit (LSB) steganography technique.

03

Dual Interface Design

Provide both a Command-Line Interface (CLI) and a Graphical User Interface (GUI) for diverse user needs.

04

Comprehensive Operations

Enable distinct 'hide', 'extract', and 'detect' functionalities within the tool.

05

User-Centric Design

Create a tool that is simple, efficient, and intuitive for all users, regardless of technical proficiency.

The LSB Advantage: How STEG-X Works

STEG-X's core functionality relies on manipulating the Least Significant Bits (LSB) of pixel values within digital images. PNG format is specifically chosen due to its lossless compression, ensuring that no pixel data is discarded, which is crucial for the integrity of embedded information.

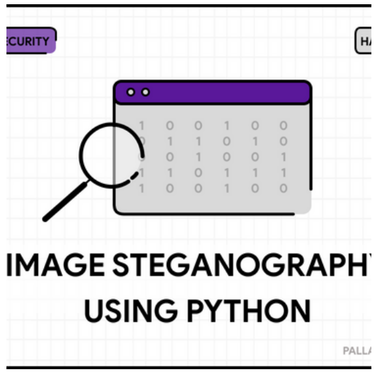
The process is meticulous: each byte of a secret message is broken down into its binary representation. These binary digits then replace the LSBs of selected image pixels. This subtle modification is imperceptible to the human eye, preserving the visual quality of the cover image while concealing vital data.

Technology Stack: Powering STEG-X

The development of STEG-X leveraged a robust set of Python libraries and tools, each chosen for its efficiency and capability in handling specific aspects of the project. This curated stack ensured both the performance and the versatility of our steganography tool.

Python

The foundational programming language, providing the core logic and extensive ecosystem for development.



stegano

A specialized library for LSB hiding and revealing operations, forming the heart of our steganographic engine.

PIL (Pillow)

Essential for comprehensive image handling, manipulation, and pixel-level analysis to prepare images for embedding.

argparse

Enabled the creation of a powerful and user-friendly Command-Line Interface (CLI) with structured arguments.

Tkinter

Provided the necessary tools for rapid prototyping and development of the intuitive Graphical User Interface (GUI).

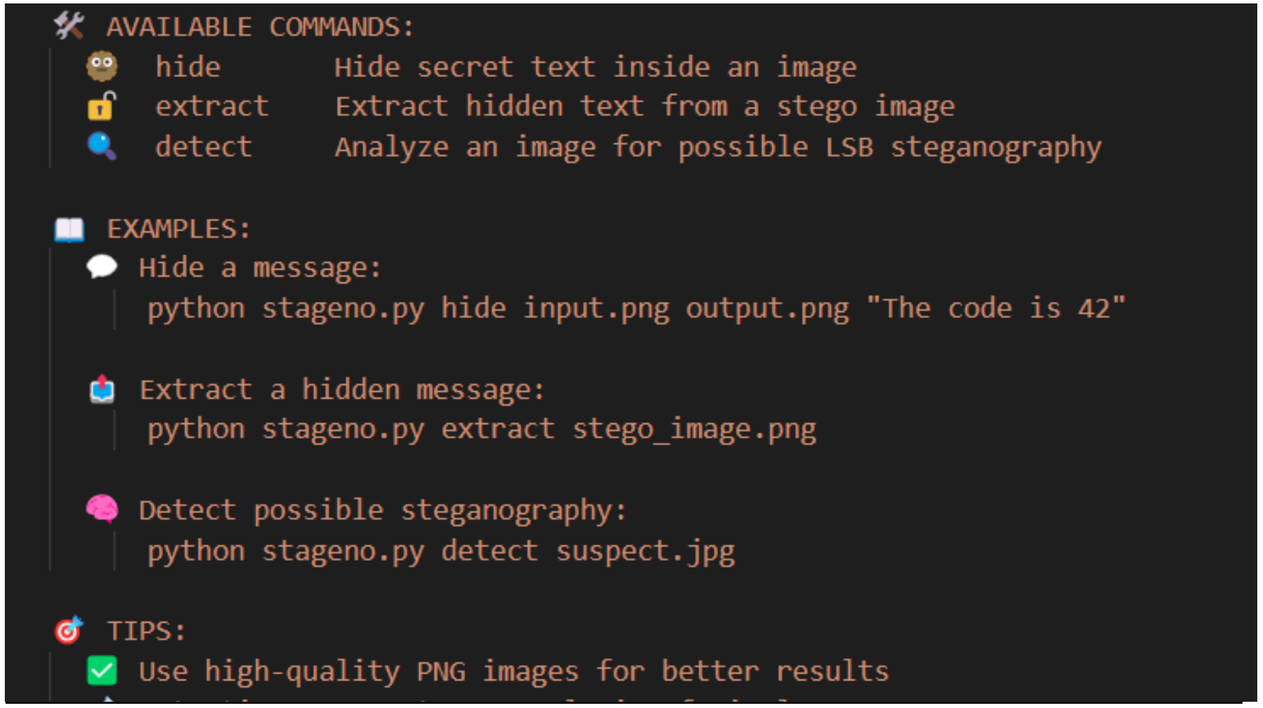
termcolor

Used to enhance the CLI experience with vibrant, colored output for better readability and user feedback.

Intuitive Interaction: CLI & GUI Design

STEG-X was designed with user accessibility in mind, offering two distinct interfaces to cater to both technical experts and general users. This dual approach ensures flexibility and ease of use across different proficiency levels.

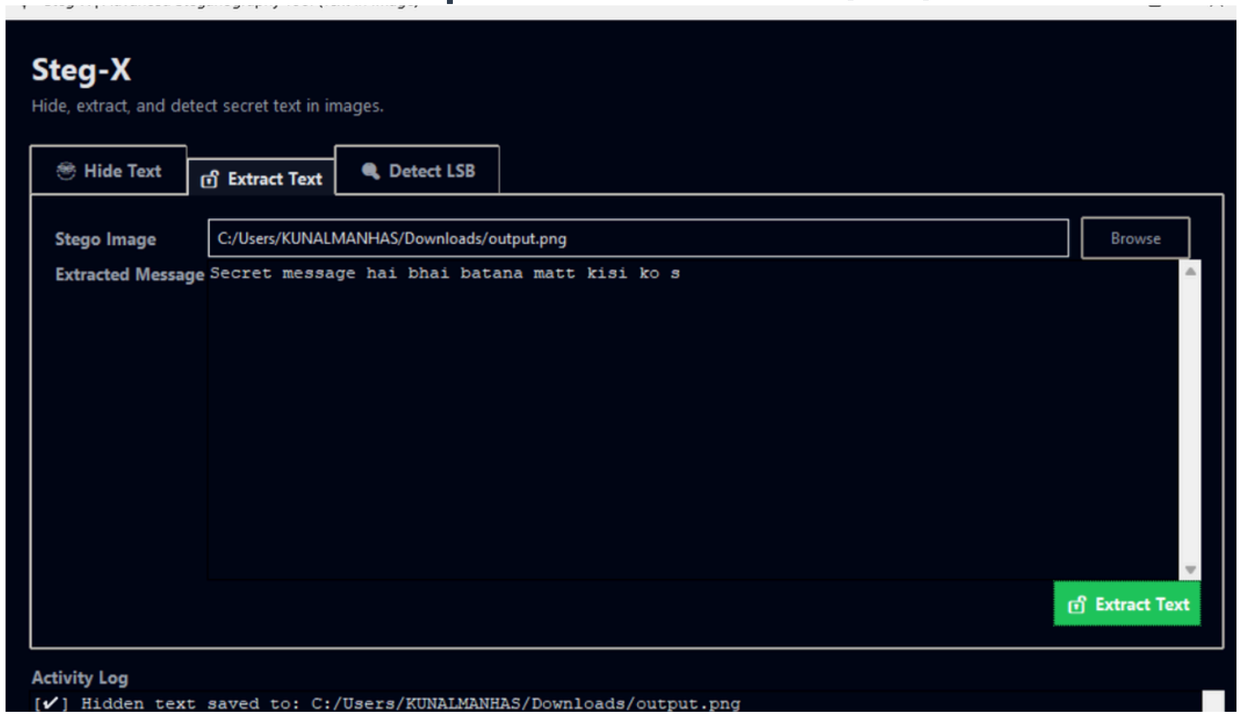
Command-Line Interface (CLI)



Developed using Python's argparse module, the CLI provides precise control for users comfortable with terminal environments. It features:

- **Three Subcommands:** hide, extract, and detect for specific operations.
- **Structured Arguments:** Clear parameters for inputs and outputs.
- **Help Menus:** Comprehensive documentation accessible directly from the command line.
- **Error Handling:** Robust mechanisms for identifying and addressing user input issues.

Graphical User Interface (GUI)



The GUI, built with tkinter, makes STEG-X accessible to non-technical users, abstracting complex commands into simple visual interactions. Key features include:

- **Action Buttons:** Dedicated buttons for 'Hide', 'Extract', and 'Detect' functions.
- **File Selection:** Intuitive dialogs for easy selection of image and text files.
- **Input/Output Boxes:** Dedicated areas for text entry and displaying results.
- **Status Messages:** Informative pop-up messages to guide users through operations.

Both interfaces are powered by the same backend logic, ensuring consistent and reliable performance.

Key Features: Comprehensive Steganography Capabilities

STEG-X is equipped with a suite of features designed to make digital steganography both effective and user-friendly. From concealing sensitive data to verifying image integrity, every function is engineered for reliability.

Message Embedding

Seamlessly hide arbitrary text messages within PNG image files, utilizing the LSB technique.

Accurate Extraction

Retrieve hidden text with precision, ensuring the complete and unaltered recovery of embedded data.

LSB Pattern Detection

Scan images for suspicious LSB manipulations, providing a mechanism for steganography detection.

PNG Format Validation

Automated checks to ensure that only compatible PNG images are used, preventing data corruption.

Dual Interface Support

Offers both Command-Line (CLI) and Graphical User Interface (GUI) options for varied user preferences.

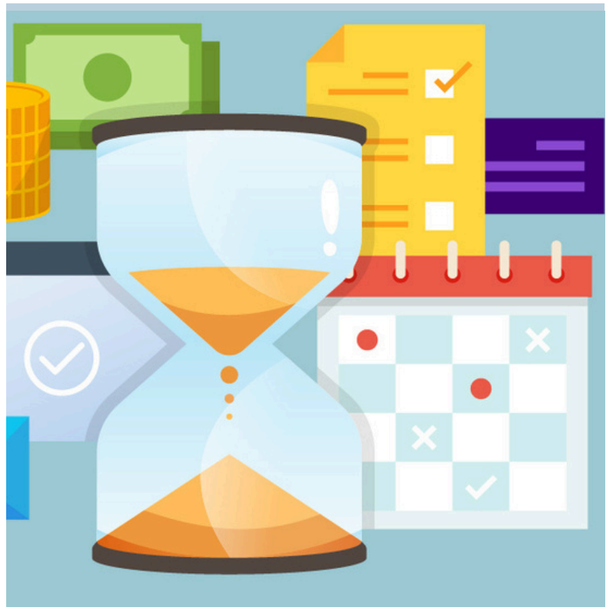
User-Friendly Workflow

Designed for intuitive operation, guiding users through each step of the hiding, extraction, or detection process.

Results & Key Learnings: From Concept to Functional Tool

The development of STEG-X culminated in a fully functional steganography tool that successfully performed all intended operations. This project was not only about building a tool but also about a significant learning experience for the developer.

Project Success



- **Message Hiding:** Successfully embedded text into PNG images without visible degradation.
- **Data Extraction:** Accurately retrieved hidden messages from stego images.
- **Pattern Detection:** Demonstrated the ability to identify LSB alterations.
- **Dual Interface:** Both CLI and GUI proved to be effective and user-friendly.

Developer Insights

- **Cybersecurity Concepts:** Deepened understanding of steganography principles and their application.
- **Python Proficiency:** Enhanced skills in Python programming, particularly in image processing and system interaction.
- **Image Processing:** Gained hands-on experience with pixel manipulation and image format intricacies.
- **Secure Tool Development:** Learned best practices for building robust and reliable cybersecurity utilities.

This project provided invaluable practical experience, bridging theoretical knowledge with real-world application in cybersecurity.

Conclusion & Future Horizons

STEG-X stands as a testament to effective cybersecurity tool development, offering a complete solution for digital information hiding. This project highlights the power of Python in creating secure, accessible, and functional applications.

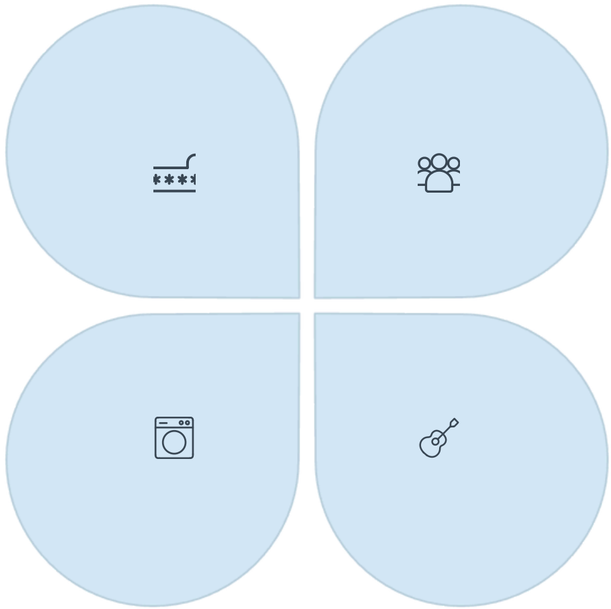
Thank You!

Enhanced Security

Integrate AES/RSA encryption for an additional layer of security before LSB embedding.

ML-Based Detection

Implement machine learning algorithms for more advanced and robust steganography detection.



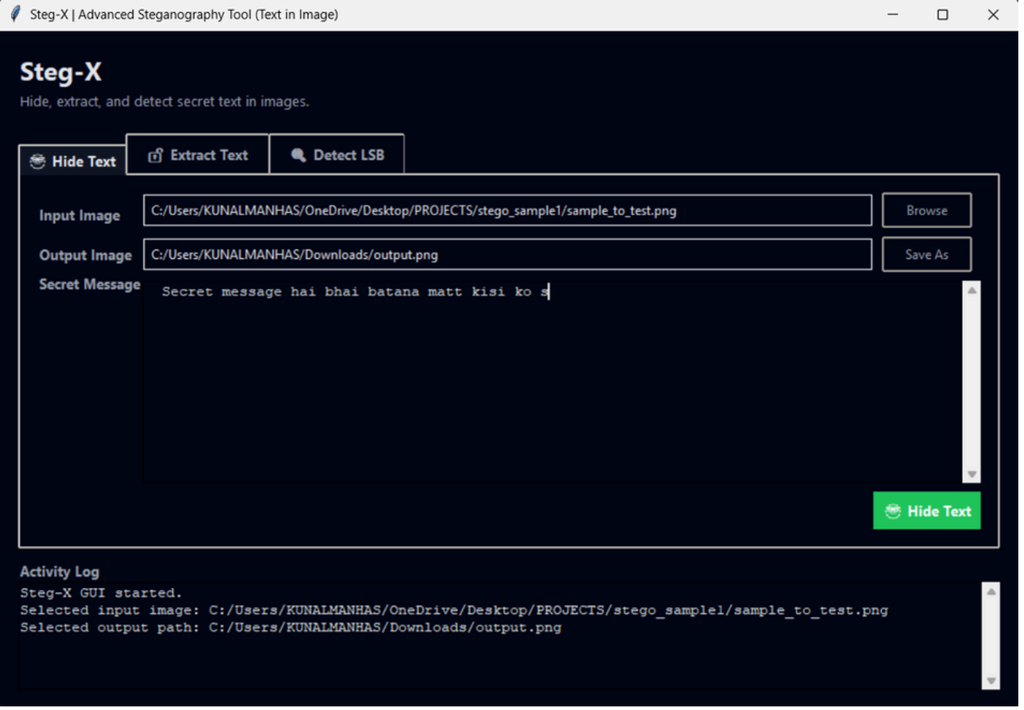
Broader Format Support

Expand compatibility to include other media types like BMP, JPG, audio, and video files.

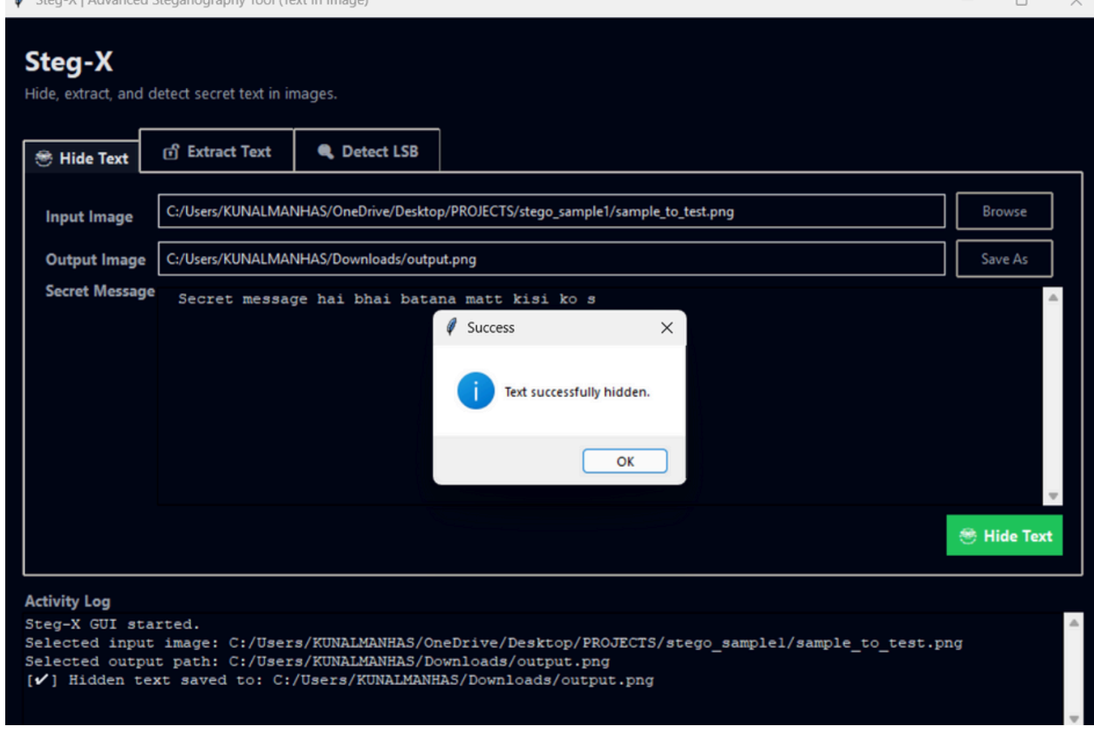
Advanced GUI

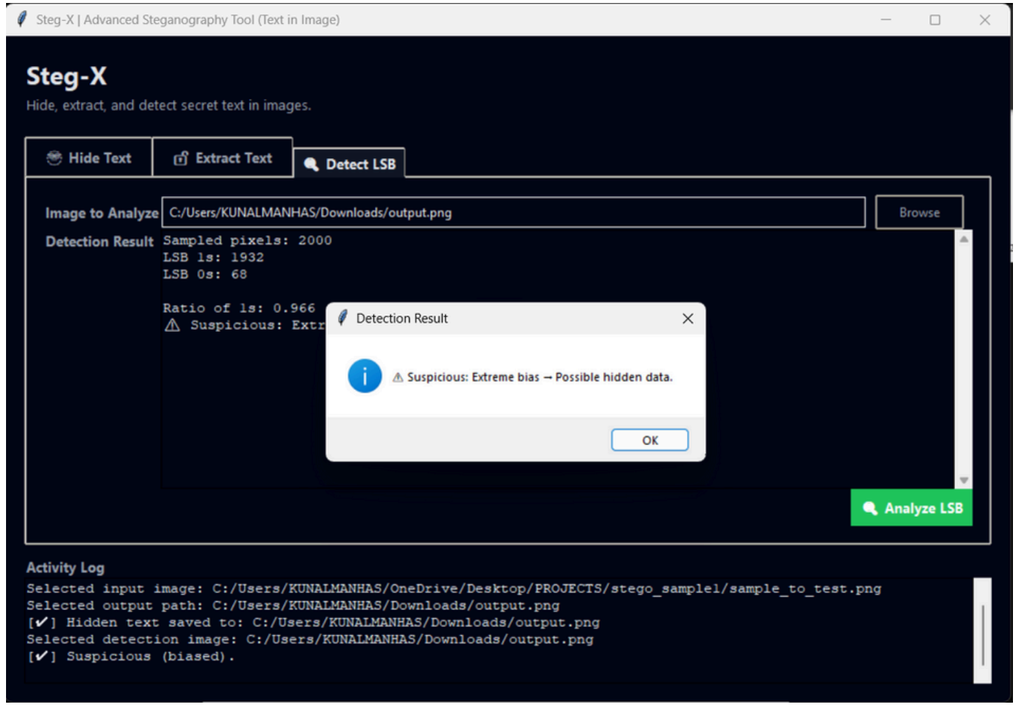
Develop a more sophisticated and feature-rich graphical interface using frameworks like PyQt.

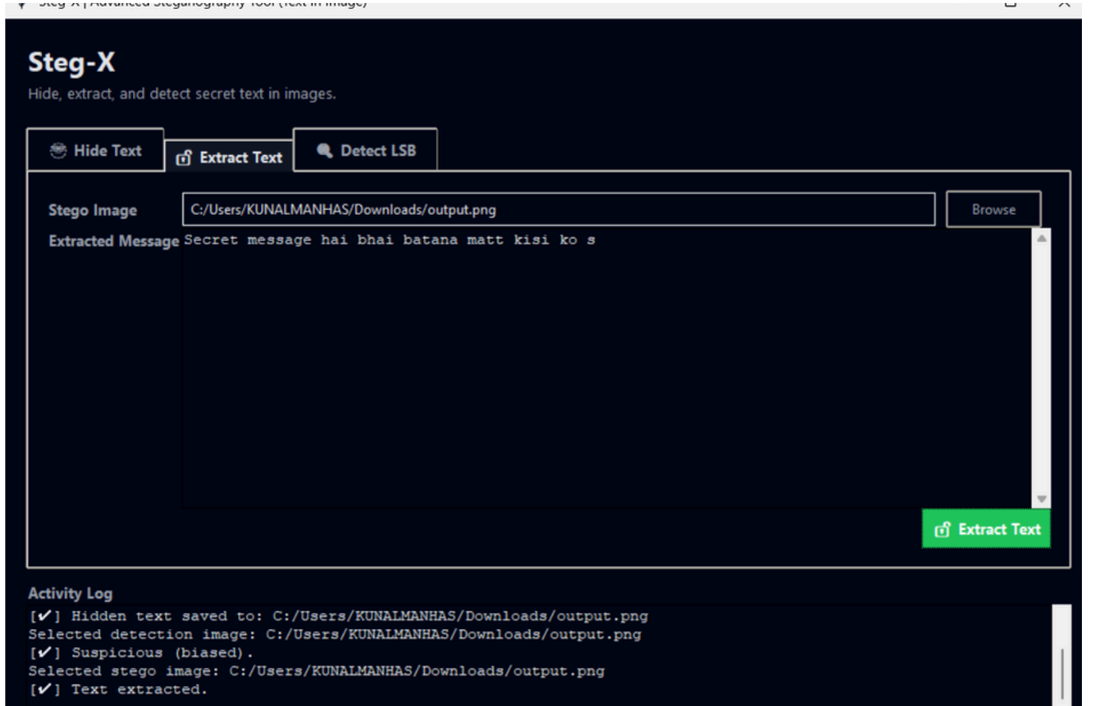
We envision STEG-X evolving into an even more versatile and resilient tool, capable of addressing the ever-changing landscape of digital covert communication. Thank you for your attention.



Sample Output Images







THANK You!!