**3. PROPOSED SYSTEM**

**3.1 Architecture**

The system proposed here is steganography, which is a method of hiding secret messages in digital media like images, audio, and video files. The design is a process-based architecture in which a cover file (X) and a secret message (M) are encoded by a steganographic encoder with a key (K) to create a stego object. This stego object is sent via a communication channel. The receiver employs a steganographic decoder, in combination with the key, to retrieve the concealed message from the stego object.  
The architecture follows these key steps:  
1. Encoding Phase: The system accepts a media file (image, audio, or video) as a cover file and inserts the secret message with the help of an encoding algorithm.  
2.Decoding Phase: The recipient uses the decoding algorithm and the key to extract the concealed message from the

stego object.

A diagram of a secret message

AI-generated content may be incorrect.

**3.2 Modules**

The system consists of several key modules that work together to enable multimedia steganography. First up is the User Interface Module, which provides a user-friendly and interactive platform for individuals to upload their media files, input secret messages, and view the results after encoding or decoding. This module ensures a smooth experience for both embedding and extracting messages. Next, we have the Encoding Module, responsible for embedding the hidden message into the chosen cover file, whether it’s an image, audio, or video. It employs steganographic techniques to hide the message within the media file while keeping any perceptual distortion to a minimum. Once the encoding is complete, the system generates a stego object that contains the concealed message. Then there’s the Decoding Module, which helps retrieve the embedded message from the stego object. Users provide the encoded media file, and the system processes it using the decoding algorithm and, if needed, an optional key. The recovered message is then displayed to the user, ensuring the successful retrieval of the hidden information. The File Handling Module takes care of uploading, processing, and storing media files throughout the encoding and decoding stages. It also ensures the safe management of stego objects, protecting against data loss or corruption during operations. Finally, the Security and Key Management Module oversees the management of encryption and decryption keys. This module enhances the confidentiality of hidden messages by safeguarding the encoding and decoding processes, ensuring that only authorized users can access the concealed information. Together, these modules create a well-organized and secure steganography system, allowing users to safely embed and extract hidden messages from digital media.

**3.3 Algorithms**

The system employs different steganographic techniques based on the type of media file used for hiding the secret message. This algorithm ensure that the hidden message remains undetectable while maintaining the quality of the cover media.

**Least Significant Bit (LSB) Algorithm**Least Significant Bit (LSB) is one of the most common image and video steganography methods. It inserts covert messages by adjusting the least significant bits of the pixel value such that any modification is negligible to the naked eye. The reason why least significant bits are minimal in the total appearance of an image or video frame is that they make insignificant contributions. For this reason, their change would not necessarily compromise visual quality.  
This technique is easy and effective to conceal data inside digital media. Yet, it is not perfect. LSB-based steganography can be attacked using steganalysis methods, wherein attackers can identify patterns of concealed data. Compression (JPEG for images, MP4 for videos) and addition of noise could also scatter or erase the hidden message. In spite of this, LSB steganography is a widely used method for secure low-complexity data concealment in multimedia data.

**4. Implementation**

**4.1 Technologies Used**

The system is built using Python as the main programming language, which provides flexibility, strong libraries, and high support for multimedia processing. Flask is used as the web framework, which provides an easy-to-use interface for users to interact directly with the encoding and decoding operations. It supports the simple handling of HTTP requests, file uploads, and rendering of templates in order to provide a smooth user interface.

For processing image and video, the system incorporates OpenCV, a useful computer vision library, to handle frames, extract visual information, and change pixel values for steganographic embedding. NumPy is utilized for numerical computation, facilitating effective array operations necessary for multimedia file processing. Pillow (PIL) is employed for image file processing, especially for text-based steganography, where text messages are hidden within images. The Stepic library also enables encoding text messages in images by modifying pixel values.

For applying Least Significant Bit (LSB) steganography, the system uses the Stegano library, which enables messages to be concealed in images and video frames without impairing visual quality. For audio steganography, the Wave module is employed to handle WAV files, adjusting the least significant bits of audio samples to encode messages without degrading sound quality.

The system uses file-based storage to efficiently handle uploaded and processed media files. This provides for seamless access to encrypted and decrypted files, making it easy to retrieve and process. Through these technologies, the system efficiently hides messages in multimedia files without compromising security, reliability, and usability.

**4.2 Algorithm**

**Algorithm for Least Significant Bit (LSB) Steganography**

The system mainly uses the Least Significant Bit (LSB) steganography method to hide and retrieve secret messages from images, videos, and audio files. Below are the algorithms for encoding and decoding messages with LSB steganography in step-by-step procedure.

**Algorithm-1: LSB Encoding (Hiding a Message)**

Input: Cover media file (image/audio/video), Secret message

Output: Stego media file with secret message

1. Convert the secret message into binary form (ASCII to binary).

2. Open the cover file (image/audio/video) and read pixel or sample data.

3. Alter the least significant bit (LSB) of every pixel (for image/video) or audio sample to insert the binary message.

4. Keep inserting the message until all bits are concealed in the cover file.

5. Store the altered media file as the stego file with the secret message.

**Algorithm-2: LSB Decoding (Retrieving a Message)**

Input: Stego media file

Output: Retrieved secret message

1. Open the stego media file and retrieve pixel or sample data.

2. Extract the least significant bit (LSB) of each pixel (for image/video) or audio sample.

3. Reconstruct the binary message by concatenating the extracted bits.

4. Convert the binary data back to text (binary to ASCII translation).

5. Display the extracted secret message to the user.

**5. Results and Discussions**

**5.1 Results**

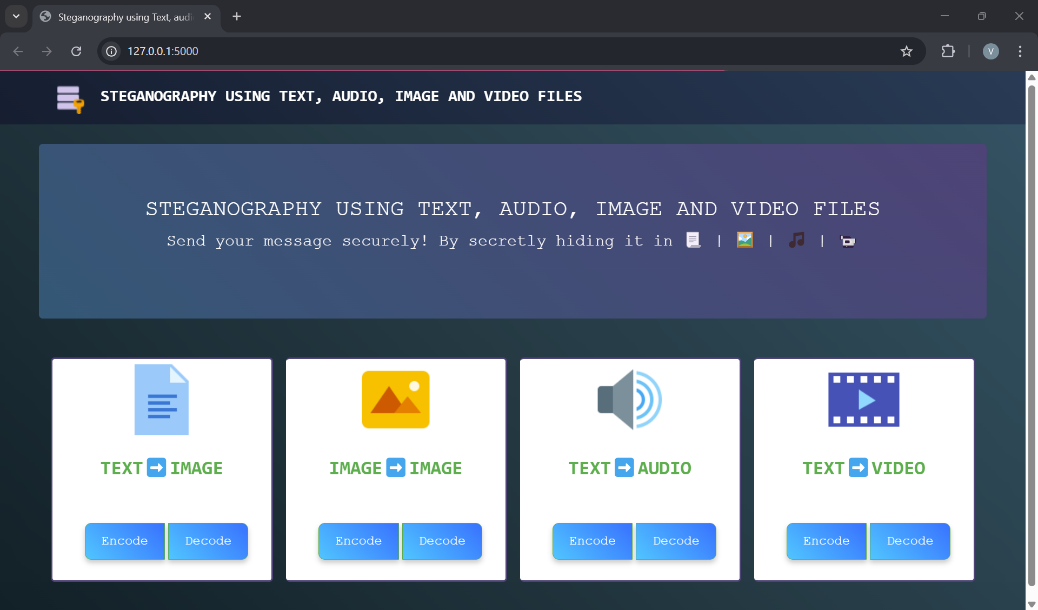
****

Image 5.1 Home page

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Conclusions and Future Enhancements**

**6.1 Conclusion**

Steganography is a widely utilized technique for data security, serving as a method to hide written text or plain text by embedding it within images. To address the challenges associated with image signal breakage, we propose an upgraded variant of steganography that offers enhanced security against Kaiseki and Friedman attacks.

The proposed technique significantly improves resistance to cryptanalysis, frequency analysis, pattern prediction, and brute-force attacks by employing multiple encryption tables. This approach introduces a high degree of diffusion and confusion in the algorithm, making the system robust and extremely difficult to compromise.

Although there are numerous steganography and cryptographic methods currently available, this field still requires dedicated research efforts to enhance data security.

**6.2 Future Enhancements**

Our future objective is to validate the proposed approach by performing comprehensive security and performance analyses. Additionally, we plan to integrate a wavelength transformation algorithm to improve accuracy in data embedding and extraction processes. This enhancement aims to further strengthen the robustness and reliability of the steganographic method, ensuring a higher level of data security.

**References**

**[1] Chen, L., & Xu, D. (2023). Secure Data Hiding in Digital Images Using LSB and Cryptographic Algorithms. IGI Global.**

**[2] Kessler, G. C. (2023). Introduction to Steganography: Methods and Applications (2nd Edition). Springer.**

**[3] Ravi, S., & Kumar, P. (2022). Data Hiding Techniques: Fundamentals and Advances in Steganography and Watermarking. CRC Press.**

**[4] Rana, N., & Sharma, M. (2023). Steganography Based on LSB with Error Correction Mechanisms in Lossless Image Formats. Journal of Information Security and Applications, 68, 103453.**

**[5] Ali, S., & Ahmad, S. (2023). Enhanced LSB Steganography Technique with Adaptive Pixel Selection for Secure Data Hiding in Images. IEEE Access, 11, 45212-45225.**

**[6] Patil, P., & Patil, S. (2022). A Novel Approach to Image Steganography Using Deep Learning and LSB Techniques. International Journal of Computer Applications, 975, 8887.**

**[7]** [**https://www.geeksforgeeks.org/what-is-steganography/**](https://www.geeksforgeeks.org/what-is-steganography/)

**[8] Transposition ciphers, columnar transposition Retrieved from** [**http://en.wikipedia.org/wiki/Transposition\_cipher**](http://en.wikipedia.org/wiki/Transposition_cipher)

**[9]** [**https://pypi.org/project/stegano/**](https://pypi.org/project/stegano/)

**[10] https://www.sciencedirect.com/science/article/pii/S1877050919313657**