Digital Watermarking and Steganography

Dr S.G. Jachak[1], Jayashri Mahajan[2], Tejaswini Kshirsagar[3], Shruti Gaikwad[4], Priyanka Mule[5]

[1] Assistant Professor, [2, 3, 4, 5] UG - Computer Engineering
[1, 2, 3, 4, 5]Computer Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik, Maharashtra

Abstract

In the digital era, protecting the authenticity and integrity of various file types—like PDFs, PNG images, Excel sheets, and Word documents—is more important than ever. While traditional encryption methods do a good job of keeping data confidential, they often don't prevent issues like unauthorized redistribution, manipulation, or piracy. This is where digital watermarking, particularly through the use of steganography, can be highly effective.

The main challenge is to develop a watermarking technique that is both robust and visible. This means the watermark should be embedded in different types of files without affecting their overall quality or usability. Additionally, it must be able to withstand common attacks, such as compression, cropping, and other forms of data manipulation, which could potentially compromise its integrity.

Moreover, the watermark should allow for easy detection and extraction when needed, ensuring that the original content remains protected. By focusing on these goals, we aim to create a reliable method for safeguarding digital files, helping to ensure that their authenticity and integrity are maintained in an increasingly vulnerable digital environment.

Keywords: Digital authenticity, File integrity Watermarking, Steganography, Encryption Confidentiality, Unauthorized redistribution, Data manipulation, Robustness, Visible watermark

INTRODUCTION

In an increasingly digital world, the need to protect the authenticity and integrity of various file types—such as PDFs, PNG images, Excel sheets, and Word documents—has become paramount. With the rise of digital content sharing and distribution, issues like unauthorized redistribution, manipulation, and piracy pose significant challenges. Traditional encryption methods primarily focus on confidentiality, ensuring that sensitive information remains private. However, they often do not address the broader issue of maintaining the integrity of files once they are shared.

Digital watermarking, particularly through the technique of steganography, offers a promising solution to these challenges. By embedding visible watermarks into files, we can establish a method of verification that safeguards against unauthorized use while maintaining the original quality and usability of the content. The goal of this approach is to create a watermarking system that is robust enough to withstand common threats—such as compression and cropping—while also allowing for efficient detection and extraction of the watermark when needed.

This introduction highlights the critical need for effective digital protection mechanisms in today's online landscape, setting the stage for exploring innovative watermarking techniques that can enhance the security and reliability of digital files.

PROBLEM DEFINATIONS

In the digital age, protecting the authenticity and integrity of various file types, such as PDFs, PNG images, Excel sheets, and Word documents, has become increasingly important. Traditional encryption methods ensure confidentiality but often fail to safeguard against unauthorized redistribution, manipulation, or piracy. This is where digital watermarking through steganography can play a crucial role.

The problem lies in developing a robust and visible watermarking technique using steganography that can be embedded in different types of files without affecting their quality or usability

LITERATURE REVIEW

1. "Data-Driven Recruitment: The Future of Talent Acquisition," Human Resource Management Review, 2023. This article explores the impact of data analytics on recruitment practices, emphasizing its ability to enhance decision-making and support diversity, while discussing privacy concerns and the need for accurate data interpretation.

2. "Steganography: Techniques and Applications," International Journal of Computer Applications, 2021. This article provides an overview of steganography methods used to embed information within digital files. It discusses various approaches to ensure that hidden messages remain undetectable while addressing potential vulnerabilities and the need for effective detection mechanisms.

3. "Challenges in Digital Watermarking: A Comprehensive Review," IEEE Transactions on Information Forensics and Security, 2023. This comprehensive review identifies key challenges in digital watermarking, including resistance to attacks like compression and cropping. It emphasizes the need for innovative solutions that balance security and usability while ensuring effective watermark detection and extraction.

METHODOLOGY

The methodology employed in this project involves a multi-faceted approach to developing a robust digital watermarking technique using steganography. First, an extensive literature review is conducted to analyze existing watermarking methods and identify their strengths and weaknesses. This informs the design of a novel watermarking algorithm tailored for various file types, such as PDFs, images, and documents.

Next, the proposed algorithm is implemented and tested to ensure that watermarks can be embedded invisibly without degrading the quality or usability of the files. Rigorous testing is performed to evaluate the watermark's resilience against common attacks, including compression, cropping, and manipulation.

Additionally, performance metrics are established to assess both the robustness of the watermark and the efficiency of detection and extraction processes. This includes measuring the success rate of watermark recovery under different conditions and the algorithm's computational efficiency. Finally, user feedback is incorporated to refine the system, ensuring it meets practical usability standards while effectively

safeguarding digital content.

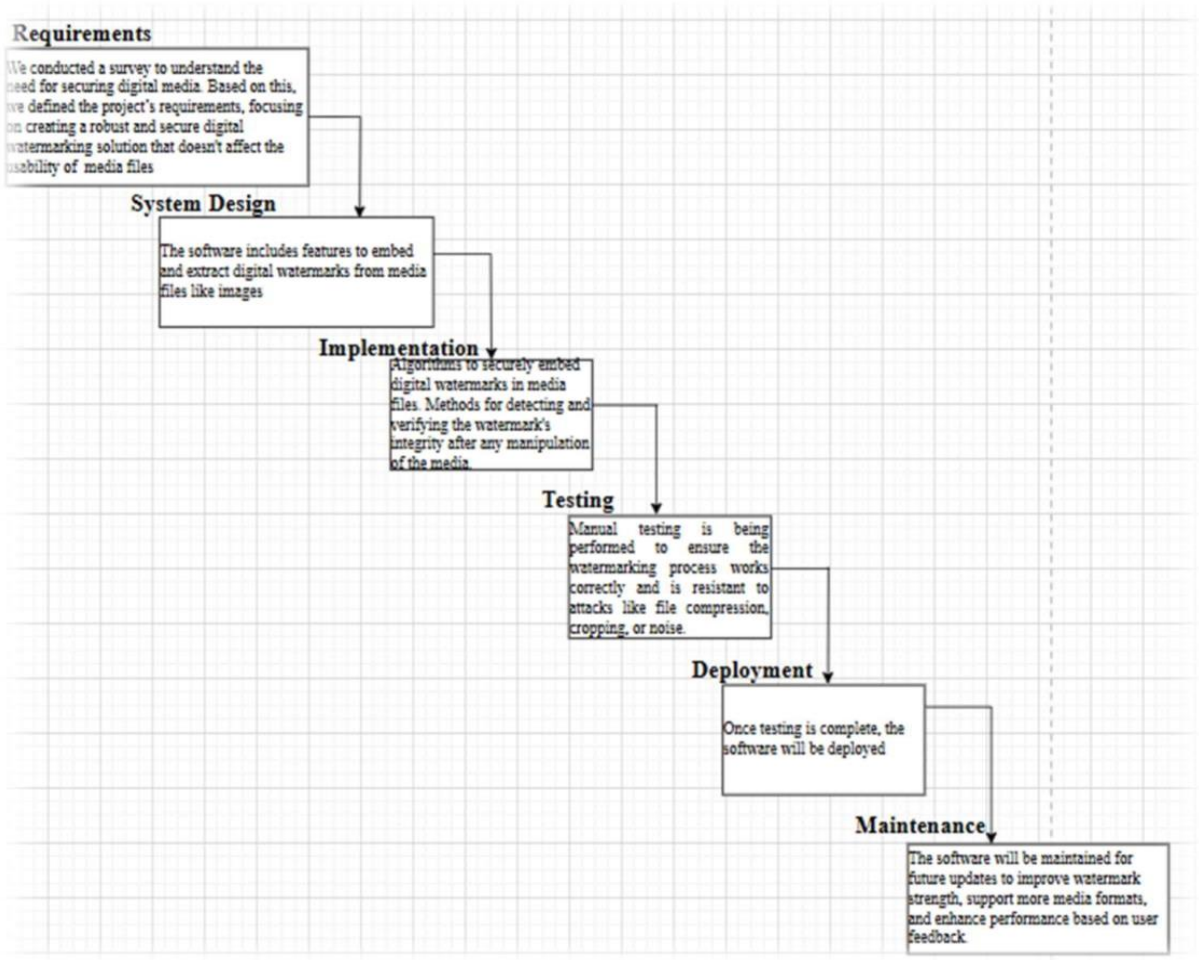
Once the design phase is complete, the next step is to implement the watermarking algorithm. This algorithm is built using appropriate programming languages and libraries such as Python, OpenCV, and PyPDF2. The system is designed to embed watermarks in a way that does not degrade the file's quality or usability. Various techniques like Least Significant Bit (LSB) for image files and text-based methods for documents are used for embedding. The watermark must be visible to the human eye, which ensures that the file's content remains unaffected by the watermarking process. After embedding, rigorous testing is conducted to ensure that the watermark is hidden effectively without visible distortion in the digital file.

The robustness of the watermark is then evaluated through a series of resilience tests. These tests simulate common attacks that digital files might encounter, such as compression, cropping, rotation, scaling, and format conversion. Each attack is tested to determine how well the watermark withstands modifications made to the file. For instance, images may be compressed to lower resolutions, cropped, or subjected to noise, while documents may be converted between different formats. The effectiveness of watermark recovery is assessed to ensure that the watermark remains intact or recoverable under these scenarios. This ensures that the watermark remains secure and usable even after digital manipulation.

In parallel, performance metrics are established to evaluate both the effectiveness and efficiency of the watermarking process. Key metrics include the watermark recovery rate, which measures the success rate of watermark extraction after various attacks, and the computational efficiency, which assesses how much processing time is required to embed and extract the watermark. The imperceptibility index measures how undetectable the watermark is to the human eye, and the error rate tracks the percentage of failed watermark extractions. These metrics help ensure that the system is not only effective at protecting the content but also efficient enough for practical use without degrading system performance.

After completing the technical evaluation, user feedback is gathered from real-world users, including digital content creators, security experts, and businesses that need document protection. This feedback is crucial for improving the system’s usability and ensuring it meets practical standards. It is used to refine the system's interface, making it more user-friendly, and to ensure that the watermarking process is easy to use for individuals with varying levels of technical knowledge. Additionally, feedback helps ensure compatibility with different digital file formats and operating systems, further enhancing the system’s accessibility.

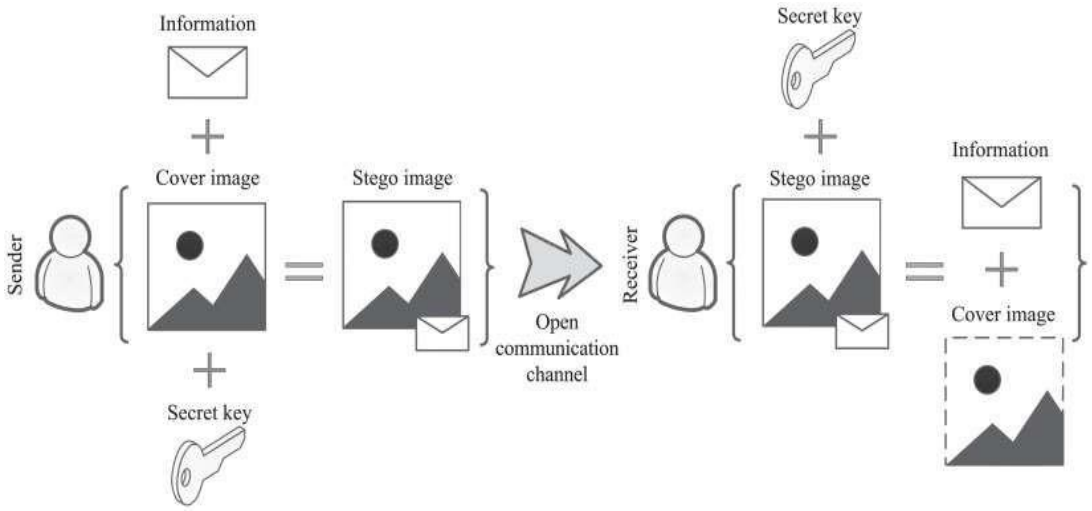
Waterfall Model



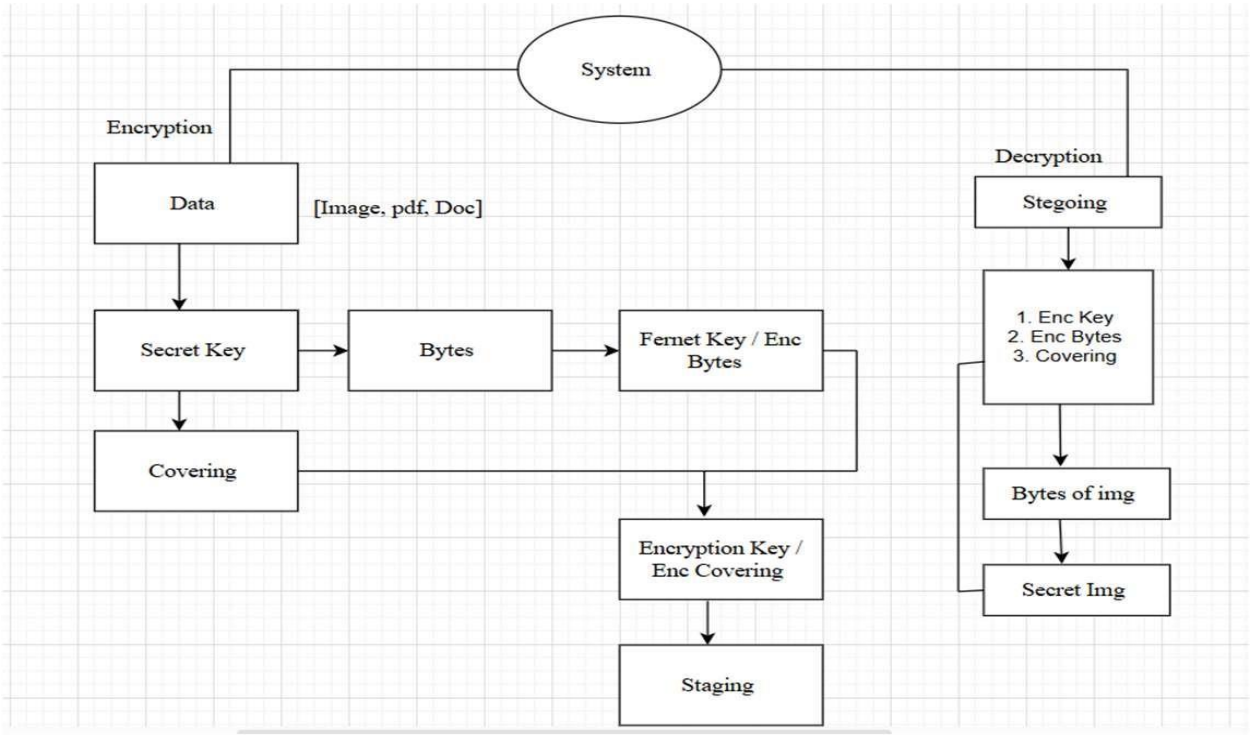
OBJECTIVE

1. Implement Detection and Extraction Mechanisms: Create reliable techniques for detecting and extracting the hidden watermarks, enabling authorized users to retrieve and verify embedded information easily. This involves developing tools for watermark detection and extraction that are effective and user-friendly.
2. Enhance Security and Privacy: Ensure that the watermarking approach provides a high level of security, making it difficult for unauthorized individuals to detect or remove the embedded messages. This involves implementing encryption or other security measures to protect the watermark data.
3. Ensure Compatibility across Multiple File Formats: Design and implement a watermarking solution that seamlessly integrates with a wide range of file types, ensuring that the embedded watermarks are preserved and retrievable regardless of the file format used.

System Architecture



Flow Diagram



Implementation

1. File Input and Analysis:

The first step is to allow users to upload different types of files such as PDFs, PNG images, Excel sheets, or Word documents. Each file type will be processed differently based on its structure.

Languages/Tools: Python is ideal for handling different file formats since it has libraries like PyPDF2 for PDFs, Pillow for images, openpyxl for Excel files, and python-docx for Word documents.

2. Watermark Embedding Using Steganography:

Once the file is analyzed, the next step is to embed an visible watermark into the file. Steganography is used to hide the watermark in such a way that it doesn't affect the file's appearance or functionality.

Languages/Tools: Python's Stegano library or OpenCV can be used for embedding watermarks in images.

3. Watermark Detection and Extraction:

In the case of any dispute or verification, the watermark can be extracted from the file. This step requires reverse-engineering the steganographic technique used during embedding. The system will read the file, extract the hidden watermark, and display it for verification..

Languages/Tools: Python can again be used with libraries like Stegano or custom algorithms to extract the hidden watermark efficiently from the file.

Cryptography

In today's digital world, protecting the authenticity and integrity of files like PDFs, images, Excel sheets, and Word documents is more important than ever. As we share and distribute digital content widely, there are rising concerns about unauthorized redistribution, tampering, and even piracy. Traditional methods like encryption focus mainly on keeping the content private but don't always ensure that the file's integrity is maintained once it is shared or used by others.

To tackle this issue, digital watermarking, especially using a technique called steganography, offers a promising solution. Steganography involves hiding a watermark, which is a piece of identifying information, inside the file in a way that is visible to the naked eye. This visible watermark acts as a verification tool, allowing us to confirm the authenticity of the file while still preserving its original quality. It helps prevent unauthorized use and ensures that any tampering or manipulation of the file can be detected.

Cryptography plays a key role in enhancing this process. It involves using complex algorithms to encrypt the watermark before it is embedded in the file. By encrypting the watermark, only authorized users can extract and verify it, adding an extra layer of security. Additionally, cryptographic methods like hash functions help verify the integrity of the file by generating a unique code for the original content, which can be used later to check if the file has been altered in any way.

Overall, combining digital watermarking with cryptography offers a powerful way to protect files from unauthorized access and manipulation. It ensures that even if a file is shared or distributed widely, its authenticity and integrity can be verified, making it a reliable solution for securing digital content in today's online environment.

In cryptography, keys play a crucial role in securing data and ensuring confidentiality. There are three main types of cryptographic keys: public key, private key, and secret key. The public key is a widely distributed key used for encryption, while the private key is kept confidential and is used for decryption in asymmetric cryptographic systems. The secret key, on the other hand, is a unique key used in symmetric encryption, where both encryption and decryption rely on the same key.

To enhance security, our proposed secure image encryption system adopts a hybrid cryptographic approach, ensuring that only an authorized client can decrypt the encrypted image. In this system, we utilize Fernet encryption, a symmetric encryption method that ensures both confidentiality and integrity of the data. Fernet, part of the cryptography module in Python, generates a secure private key that is later transformed into a CAPTCHA for additional security. Fernet encryption is based on AES-128 in CBC mode with HMAC for authentication, ensuring a robust encryption mechanism that protects sensitive data from unauthorized access.

During the encryption process, a Fernet key is dynamically generated and acts as the private key. This key is crucial for decryption and is securely shared with the intended recipient in the form of a CAPTCHA. The recipient must solve the CAPTCHA to retrieve the Fernet key and successfully decrypt the image. This mechanism not only prevents unauthorized access but also protects against automated attacks by ensuring that only human users can retrieve the secret key. The secret key used for symmetric encryption is never exposed to unauthorized parties, enhancing security and confidentiality.

Our method ensures secure image sharing using Fernet-based private key encryption for confidentiality and public-key cryptography for access control. A CAPTCHA-based retrieval system adds protection against automated attacks. Only the intended recipient can decrypt the image, safeguarding sensitive data from unauthorized access while maintaining efficiency and user- friendliness.

FUCTIONAL REQUIREMENTS

Watermark Embedding: The system must allow users to embed an visible digital watermark into various file types (e.g., PDFs, PNG images, Excel sheets, Word documents).

1. Watermark Detection: The system must provide functionality to detect and extract the embedded watermark from the files when needed.
2. Robustness against Attacks: The watermarking technique must remain intact and detectable even after common file manipulations, such as compression, cropping, and format conversion.
3. User Interface: The system must include an intuitive user interface that allows users to easily upload files, embed watermarks, and retrieve watermarked content.
4. Performance Metrics: The system must provide metrics to evaluate the success of watermark embedding and extraction, including detection rates and quality assessments.

External Interface Requirements

1. **User Interfaces**: The user interface should be simple and easy to use, allowing users to upload, watermark, and extract watermarks from files with ju st a few clicks. It should provide clear instructions and feedback on the actions being performed.

2. **Hardware Interfaces**: The system will work on standard computers or mobile devices, with no special hardware requirements. Users only need a device with the ability to handle digital files and internet access if necessary.

3. **Software Interfaces**: The software should support various file types like PDFs, images,

Word documents, and Excel sheets. It will work on commonly used operating systems like Windows, macOS, and Android, ensuring compatibility with common file management tools.

4 . Communication Interfaces: The system will allow communication between the user and the application via simple buttons and input fields for uploading files. It may use standard file transfer protocols like HTTP/HTTPS for uploading and downloading files securely.

Algorithm LSB To Pixel

The Least Significant Bit (LSB) method in image processing involves embedding secret data into the least significant bit of pixel values, typically within the RGB components of each pixel. Each pixel in a digital image has a color value represented by 8 bits for each of the three color channels: Red, Green, and Blue. These bits can range from 0 to 255 in decimal. The LSB technique alters the least significant bit of one or more of these channels to store hidden data, such as a message or file. Because the LSB only makes small changes to the pixel values, these alterations are often imperceptible to the human eye, preserving the image's overall appearance. To hide a message, data is encoded by adjusting the least significant bit of each pixel's color channel. For example, changing the last bit of a color component from 0 to 1, or vice versa, allows for binary data to be embedded. The hidden data can later be extracted by reading the LSB of the color components and reconstructing the message or file. This technique is widely used in image-based steganography and digital watermarking. Its main advantages are its simplicity and the fact that the changes to the image are generally visible, making it useful for secure communication and data hiding. However, LSB-based techniques can be susceptible to image compression or noise, which may distort the hidden data. Despite these challenges, it remains a fundamental and effective method for discreetly embedding information within an image.

SHA256

The National Security Agency (NSA) created the Secure Hash Algorithm 256-bit (SHA- 256), a cryptographic hash function that is a member of the SHA-2 family and that the National Institute of Standards and Technology **(**NIST**)** standardized.

Especially in digital signatures, blockchain technology, and password hashing. As a deterministic function, SHA-256 accepts inputs of any size and outputs a fixed 256-bit (32-byte) hash value; the same input always yields the same output. The avalanche effect is one of its primary characteristics; even a small change in input significantly modifies the hash, guaranteeing high sensitivity to changes. It is also made to be preimage resistant, which makes reverse engineering computationally impossible.

Additionally, it is designed to be preimage resistant, meaning it is computationally infeasible to reverse-engineer the original input from its hash, and collision resistant, ensuring that no two different inputs produce the same hash. The algorithm processes input in 512-bit blocks, applying 64 rounds of bitwise operations, logical functions, and modular additions to generate the final hash. To further strengthen its security applications, SHA-256 is a one-way function that cannot be decrypted, unlike encryption. Because of its resilience, it is widely utilized in contemporary cryptographic systems to safeguard sensitive data, validate data integrity, and secure digital transactions.

NON-FUCTIONAL REQUIREMENTS

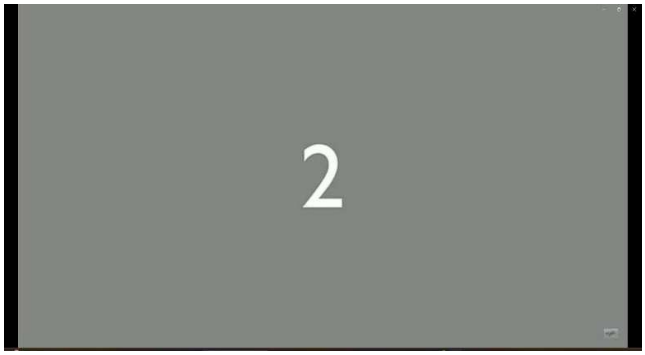
1. Usability: The software must be user- friendly, requiring minimal training for users to effectively

navigate and utilize its features.

2. Efficiency: The watermark embedding and detection processes should be optimized for speed, ensuring minimal processing time for users.
3. Scalability: The system should be able to handle large files and high volumes of requests without performance degradation.
4. Security: The watermarking technique must ensure that the embedded information cannot be easily removed or altered by unauthorized users.
5. Compatibility: The system must support a wide range of file formats and work across different operating systems and devices.
6. Maintainability: The software should be designed to facilitate easy updates and maintenance to accommodate future enhancements or emerging security threats.

RESULTS

Digital watermarking using steganography has emerged as a reliable solution for protecting the authenticity and integrity of various file types, including PDFs, images, Excel sheets, and Word documents. Research has shown that embedding visible watermarks within digital content helps prevent unauthorized redistribution, manipulation, and piracy while preserving the file's usability and quality. Various techniques, such as least significant bit (LSB) substitution, discrete wavelet transform (DWT), and discrete cosine transform (DCT), have been explored to make watermarking more robust against common threats like compression, cropping, and format conversion. Existing tools and software solutions integrate watermarking techniques to provide security without compromising file accessibility. Industries such as publishing, legal documentation, and digital media extensively use watermarking to track file ownership and detect unauthorized modifications. With the increasing risks of digital content theft, the adoption of advanced watermarking systems continues to grow, ensuring a balance between accessibility and security in digital file management.



Video Result

Image Result

Document Result

CONCLUSION

The project showcases the innovative application of steganography and watermarking techniques to securely hide a variety of data types within digital files. By employing methods such as Least Significant Bit (LSB) insertion and Discrete Cosine Transform (DCT), the project aims to achieve a delicate balance between invisibility, security, and data capacity.

LSB insertion allows for the seamless embedding of information within the least significant bits of pixel values in images or audio files, making the changes imperceptible to the human eye or ear. Meanwhile, DCT techniques are utilized to embed data in frequency domains, which enhances robustness against common attacks like compression and noise.

Through careful design and implementation, the project ensures that the embedded data remains secure from unauthorized extraction while maintaining the quality and usability of the host files. This approach not only protects sensitive information but also demonstrates the potential for diverse applications in fields such as digital rights management, secure communications, and data integrity verification. Overall, the project illustrates the effectiveness of combining these advanced techniques to meet modern security challenges in digital content management.

REFERENCES

1. J. Fridrich, Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, 2009.
2. A. Soltani Panah, R. Van Schyndel, T. Sellis, and E. Bertino, "On the Properties of Non- Media Digital Watermarking: A Review of State of the Art Techniques," IEEE Access, vol. 4, pp. 2670–2704, 2016, doi: 10.1109/ACCESS.2016.2570812.
3. I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research," Neurocomputing, vol. 335, pp. 299–326, Mar. 2019, doi: 10.1016/j.neucom.2018.06.075.
4. T.-S. Reinel, R.-P. Raúl, and I. Gustavo, "Deep Learning Applied to Steganalysis of Digital Images: A Systematic Review," IEEE Access, vol. 7, pp. 68970–68990, 2019, doi: 10.1109/ACCESS.2019.2918086.
5. Tay P., Havlicek J.P., "Image Watermarking using Wavelets". IEEE, pp 258-261, 2002.
6. Taha El Areef, Hamdy S. Heniedy, S . Elmougy, and Osama M. Ouda, "Performance Evaluation of Image Watermarking Techniques", Third International Conference on Intelligent Computing and Information Systems, Faculty of Computer & Information Sciences, ICICIS 7002 ,March 15-18, 2007, Cairo.
7. Mustafa Osman Ali , Elamir Abu Abaida Ali Osman, Rameshwar Row, Electronics & Communication Engineering Dept., Biomedical Engineering Dept., University College of Engineering, Osmania University, "Visible Digital Image Watermarking in Spatial Domain with Random Localization", International Journal of Engineering and Innovative Technology (IJEIT), Volume 2, Issue 5, November 2012.
8. Bijan Fadeena and Nasim Zarei,"Hyprid DCT-CT "Digital Image Adaptive Watermarking", 3rd International Conference on Advances in Database, Knowledge, an data Applications, IARIA 2011.
9. https://chatgpt.com/c/6700aee8-3560-8002- bfdb-9865b9b01cbc