ANDROID BASED IMAGE STEGANOGRAPHY

Amsavarthan P [1], Praveen P [2], Ajay N [3], Gopalakrishnan B [4]

[1,2,3] UG – B. Tech Information Technology, Bannari Amman Institute of Technology,

[4] Professor Information Technology, Bannari Amman Institute of Technology,

Sathyamangalam, Tamilnadu.

amsavarthan.it20@bitsathy.ac.in, praveen.it20@bitsathy.ac.in,

ajayn.it20@bitsathy.ac.in, GOLAPAKRISHNANB@bitsathy.ac.in

ABSTRACT

Android-based image steganography is a method of concealing confidential information within digital images on Android mobile devices. This technique capitalizes on the widespread use of Android platforms and the growing trend of image-based communication. It involves the embedding of sensitive data, such as text or other files, into the pixels of an image while preserving the image's visual appearance. This field employs various algorithms and methodologies to ensure both effective data hiding and minimal perceptual impact on the host image. Android-based image steganography plays a crucial role in secure communication, data protection, and privacy maintenance. It enables covert sharing of information, making it challenging for unauthorized users to detect the hidden data. Despite its significance, challenges exist, such as ensuring robustness against attacks and maintaining the balance between data capacity and image quality. Applications of this technology span from secure messaging and digital watermarking to copyright protection and authentication. In the era of mobile computing and digital communication, Androidbased image steganography emerges as an essential tool for safeguarding sensitive information within the Android ecosystem, ensuring the confidentiality and integrity of data in the face of evolving cyber threats.

KEYWORDS: *Android devices, image-based communication, data hiding, digital watermarking, authentication, copyright protection.*

1. INTODUCTION

Android-based image steganography is a cutting-edge field that marries the ubiquity of Android mobile devices with the art of concealing sensitive information within digital images. In an era where smartphones have become an extension of our daily lives, this technology plays a pivotal role in securing data, maintaining privacy, and enabling covert communication. This field's significance extends beyond the realm of personal communication. In a world where cybersecurity threats loom large, Android-based image steganography finds applications in secure messaging, digital watermarking, copyright protection, authentication, and more. It empowers individuals and organizations to shield sensitive information from prying eyes while maintaining seamless communication. Yet, Android-based image steganography faces multifaceted challenges. Striking the right balance between data capacity and perceptual alterations is a delicate task, as overembedding can degrade image quality. Robustness against detection and attacks is another concern, requiring constant innovation to stay ahead of evolving security threats. At its core, Android-based image steganography operates by embedding hidden information, such as text, files, or even other images, into the pixels of a host image without visibly altering its appearance. This covert embedding ensures that the casual observer remains unaware of the presence of concealed data, thereby providing a robust layer of security for the exchanged information. With the proliferation of image-centric communication on social media platforms and messaging apps, this approach gains immense relevance in ensuring the confidentiality of personal and corporate data. The beauty of Android-based image steganography

lies in its adaptability to the Android ecosystem.As Android devices continue to dominate the global mobile market, integrating 11 steganographic techniques into these platforms empowers users with a versatile and accessible means of protecting their data. However, achieving effective steganography on Android presents its unique set of challenges, including optimizing data capacity while minimizing perceptual alterations and addressing potential vulnerabilities to detection. This introductory exploration delves into the fascinating world of Android-based image steganography, shedding light on its importance in securing sensitive information, enabling secure communication, and preserving privacy in an increasingly interconnected and data-driven world. As technology continues to advance, this field evolves as a vital resource for individuals and organizations seeking innovative ways to protect their digital assets and maintain data confidentiality on Android-based devices. Android-based image steganography relies on the Least Significant Bit (LSB) technique to conceal confidential data within digital images. LSB substitution involves replacing the least significant bits of pixel values with hidden data, ensuring minimal visual distortion. This method is favoured for its simplicity and effectiveness. LSB steganography's data capacity depends on image size and the number of LSBs altered per pixel, with larger images allowing for greater data concealment. In colour images, each colour channel (RGB) can be manipulated independently to increase data capacity. However, LSB-based steganography is vulnerable to detection through statistical analysis and may not be robust against compression or image editing. Despite its limitations, LSB steganography is widely used in Android applications for secure messaging, watermarking, and covert data transfer. Its practicality, minimal impact on image quality, and suitability for mobile devices make it a valuable tool for safeguarding privacy and data integrity in the Android ecosystem. 12 In this introductory exploration, we delve into the captivating universe of Android-based image steganography, emphasizing its critical role in securing data, enabling secure communication, and preserving privacy within the Android ecosystem. As technology advances and mobile devices continue to shape our lives, this field stands as a beacon of innovation, offering ingenious solutions to the ever-growing challenges of data protection and confidentiality in a world increasingly reliant on Android-based mobile computing.

2. RELATED WORKS AND LITERATURE SURVEY

Android Based Image Steganography: In this study, text data is encoded at the sender behind the Image cover object and decrypted at the receiver using the application's additional security features. Since the image serves as the ideal source to conceal the message that is undetectable to the human sight, we suggest an object in this work by altering the least relevant parts of the image's pixel data. The text message in this application is encoded on the back of the sender's image card object and decoded on the recipient's device, which is also equipped with extra security protections. Steganalysis Based on Differential Statistics: This study improved Fridrich's approach for identifying steganography in grayscale JPEG images and incorporated differential statistics for revealing concealed data in grayscale raw photos. In order to represent data spatial correlations in raw images that extend to higherorder differentiations, the co-occurrence matrix, based on nearby pixel intensities, was used. For steganalysis in raw pictures, the centre of mass (COM) values of the histogram character functions (HCFs) were calculated. Differential statistics were gathered for JPEG files from the DCT block boundaries in the decompressed pictures. These statistics and DCT domain data were generated to produce 28-dimensional feature vectors for JPEG picture analysis. Digital image steganography: The science of steganography entails conveying sensitive information in a suitable multimedia carrier, such as image, audio, and video files. The premise is that if the feature is obvious, the point of attack is obvious, hence the objective is always to hide the fact that the embedded data is even present. The many steganography techniques that are now in use are reviewed and critically analysed in this study along with some general standards and best practises that have been taken from the literature. The object-oriented embedding technique is supported by the recommendations and arguments in this paper's conclusion. Though it is not the focus of this examination, steganalysis—the science of countering steganography—will be briefly covered. Android Based Secret Information Sharing Using Steganography: The many steganography techniques that are now in use are reviewed and critically analysed in this study along with some general standards and best practises that have been taken from the literature. The object-oriented embedding technique is supported by the recommendations and arguments in this paper's conclusion. Though it is not the focus of this examination, steganalysis—the science of countering steganography—will be briefly covered. Steganography and Steganalysis: An Overview: Despite having a long history, steganography is still a dynamic and developing technology that keeps up with the rapid advancement of cutting-edge digital instruments. In order to properly detect hidden information, steganalysts and their analytical tools must also keep

up with the development of steganographic techniques. It's critical to understand that steganography, like any technology, is morally neutral; the application determines its ethical standing. Depending on how it is applied, it could either help or hurt society. Steganography can improve privacy, secure communications, and preserve sensitive data 15 when used responsibly and legally. However, when used maliciously, it can significantly compromise security and privacy. A Study on Steganography Concealing Data: Steganography is a technique for preventing unauthorised access to personal information. Steganography is utilised in a variety of industries, including the military, aerospace, and business. Techniques such as steganography are used to convey data securely from sender to recipient. The computer programmers start steganographic methods on both audio and video files. Due of the prevalence of sending digital photos via email and other forms of Internet communication, steganography systems are now using multimedia data, such as images, audio, and video, as a cover medium. This review's objective is to introduce young scholars to the concept of steganography.

3. OBJECTIVES

The objective of Android-based image steganography is to develop and implement efficient and secure techniques for concealing confidential data within digital images on Android mobile devices.

3.1 KEY COMPONENTS AND TASKS

This field aims to: Enable covert communication: To provide a means for individuals and organizations to exchange sensitive information discreetly through images in the Android ecosystem. Preserve data integrity: To embed hidden data within images while maintaining the visual quality and integrity of the host image. Enhance privacy protection: To safeguard personal and sensitive data from unauthorized access and maintain user privacy on Android devices. Facilitate secure messaging: To offer a secure channel for messaging and data sharing, particularly in environments where data security is paramount. Counter steganalysis: To develop methods that withstand steganalysis techniques and remain undetectable by unauthorized parties. Explore novel algorithms: To continuously research and innovate steganographic algorithms that can adapt to the evolving landscape of Android technology. Address vulnerabilities: To identify and mitigate potential weaknesses in Android-based steganography, ensuring robustness against various attacks. Promote versatile applications: To encourage the use ofsteganography for purposes such as digital watermarking, copyright protection, and authentication within the Android ecosystem. Educate users: To raise awareness about the importance of data protection and privacy preservation through Android-based image steganography techniques. Keep pace with technological advancements: To stay up-to-date with the latest developments in mobile computing, image processing, and security, ensuring the continued relevance and effectiveness of steganographic methods on Android devices.

3.2 METHODOLOGY

The methodology used in Android-based image steganography involves several key steps: Image Selection: Image selection in Android-based image steganography is a critical step that involves purposefully choosing a host image with careful consideration of several factors. First and foremost, the selection should align with the intended purpose, ensuring that the chosen image is contextually relevant to the communication or environment where it will be shared. Additionally, the image's size is of utmost importance, as it must possess sufficient capacity to conceal the desired data without causing noticeable distortion to the image's quality. Content relevance is another key aspect, ensuring that the host image's subject matter corresponds with the hidden data to minimize suspicion. Compatibility with Android devices, in terms of format like JPEG or PNG, is essential for seamless processing. The image's visual characteristics, such as color, texture, and complexity, must be evaluated to assess its ability to effectively hide alterations introduced during data embedding. Data Encoding: Data encoding in the context of Android-based image steganography involves the conversion of sensitive or confidential information, such as text, files, or messages, into a format that can be discreetly concealed within the pixels of a host image. Typically, this process converts the data into binary code, representing each piece of information as a series of ones and zeros. The binary data is then integrated into the host image using steganographic techniques like Least Significant Bit (LSB) substitution. This ensures that the concealed data blends seamlessly with the image's pixel values, making it challenging for unauthorized users to detect. The choice of encoding method depends on the type of data and the steganographic algorithm in use. For instance, text messages may be converted into binary ASCII code, while files are broken down into binary bytes. The key objective of data encoding is to preserve the original data's integrity while making it compatible for hidden

transmission or storage within an image. Successful data encoding is fundamental to the confidentiality and security of sensitive information, as it enables covert communication while maintaining the visual integrity of the host image. LSB Embedding: Least Significant Bit (LSB) embedding is a widely-used technique in image steganography, including Android-based steganography. It involves concealing data within a digital image by subtly altering the least significant bits of the pixel values. In this method, binary data, often representing confidential information, is encoded into the least significant bit of each pixel's color channel (Red, Green, and Blue in color images), one bit at a time. Since the least significant bits contribute the least to the overall pixel value, these small alterations are typically imperceptible to the human eye, preserving the visual quality of the image. LSB embedding can be applied to individual color channels or grayscale images, allowing for the hidden data to be seamlessly integrated into the image, making it a popular choice for covert communication and data protection. Encryption (Optional): Encryption is a fundamental security process that involves transforming plaintext or readable data into ciphertext or an unreadable format using cryptographic algorithms and keys. This transformation ensures data confidentiality and security by rendering it unintelligible to unauthorized individuals or entities. During encryption, the original data is combined with an encryption key, generating ciphertext that appears as random and nonsensical characters. To decrypt and access the original data, the recipient must possess the corresponding decryption key, which reverses the encryption process. Advanced encryption methods employ complex mathematical algorithms, providing a high degree of data protection and security. Encryption is essential in safeguarding sensitive information, such as financial transactions, personal communications, and confidential documents, particularly in an era where digital privacy and security are paramount. Advanced encryption methods employ complex mathematical algorithms, ensuring a high degree of data protection in today's interconnected digital world. Steganalysis Resistance: Steganalysis Resistance refers to the capacity of a steganographic technique, particularly in the context of Android-based image steganography, to withstand attempts at detection by steganalysis, which is the process of identifying hidden data within an image. To achieve steganalysis resistance, methods and strategies are employed to minimize the detectable traces left by the hidden data. These measures can include embedding the data in less conspicuous areas of the image, adding subtle alterations that mimic natural image noise, and using sophisticated embedding algorithms that make it challenging for steganalysts to discern the presence of hidden data. Furthermore, steganalysis-resistant techniques often adapt to evolving steganalysis methods, ensuring that even as detection techniques advance, the concealed data remains undetectable. This is critical for maintaining the security and confidentiality of the hidden information, especially in scenarios where sensitive data is at stake. By continuously enhancing steganalysis resistance, Android-based image steganography methods can reliably protect confidential information and maintain their effectiveness in an everchanging landscape of digital security challenges. Image Saving: Image saving, in the context of Android-based image steganography, is the process of preserving the digital image that has been steganographically modified to conceal hidden data. Once the data embedding is complete, the modified image is saved onto the Android device's storage or transmitted to another recipient. It is crucial to ensure that the saved image maintains its visual quality and integrity while concealing the hidden data. Care must be taken during this step to choose an appropriate file format and compression settings to prevent unintended exposure of the concealed information or degradation of image quality. The saved image becomes the carrier for the hidden data and can be used for secure communication or storage, with the expectation that the concealed information remains confidential and accessible only to authorized users with knowledge of the steganographic technique used for embedding. Data Extraction: Data extraction in the context of Android-based image steganography is the process of retrieving the concealed or hidden information from a steganographically modified image. This operation typically involves using the same steganographic algorithm and key that were initially used for data embedding. During extraction, the steganographic algorithm scans the modified image, identifies the hidden data bits, and reconstructs the original information. Once extracted, the data can be decoded and made accessible for its intended purpose, such as reading a hidden message or accessing confidential files. Data extraction is a critical step in steganography, as it ensures that the concealed information remains retrievable by authorized parties while maintaining the security and confidentiality of the hidden data against unauthorized access. Error Checking: Error checking, in the context of Android-based image steganography, involves the implementation of mechanisms to verify the integrity and accuracy of the extracted hidden data during the data extraction process. These mechanisms are essential to detect any potential corruption or errors that may have occurred during the data embedding and extraction phases. Common error checking techniques include the use of checksums, cyclic redundancy checks (CRC), or cryptographic hashes, which generate unique values or signatures for the original data. During extraction, these values are compared to the extracted data, and any discrepancies or mismatches indicate potential errors or tampering. Error checking ensures that the concealed information remains intact and unaltered, enhancing the reliability and security of the steganographic process,

especially in scenarios where data integrity is critical. Security Measures: Security measures in Android-based image steganography encompass a range of strategies and safeguards employed to protect the confidentiality and integrity of hidden data. These measures may include password protection, encryption of the concealed information, and the use of encryption keys. Password protection restricts access to the hidden data, ensuring that only individuals with the correct password can extract and decode it. Encryption adds an extra layer of security by encoding the hidden data in such a way that even if the steganographically modified image is accessed, the concealed information remains unreadable without the decryption key. These security measures are essential for safeguarding sensitive data and ensuring that only authorized users can access and decipher the concealed information, providing a robust defense against unauthorized access or tampering. Evaluation and Testing: Evaluation and testing in Android-based image steganography are crucial steps to assess the effectiveness, reliability, and security of the steganographic techniques employed. During evaluation, the steganographic method is subjected to a battery of tests, including steganalysis, to determine how well it conceals hidden data and whether it can withstand detection attempts. This process helps identify vulnerabilities and areas for improvement. Additionally, ethical considerations and legal compliance are addressed to ensure that the steganographic techniques adhere to privacy and copyright laws. Evaluation and testing are iterative processes, driving ongoing refinement and innovation in Android-based image steganography, ultimately enhancing the reliability and security of data concealment methods in an ever-evolving digital landscape.

4. PROPOSED WORK MODULES

There are several proposed modules for Android for image steganography. Some of the most common ones include: DCT (Discrete Cosine Transform) embedding: DCT (Discrete Cosine Transform) embedding is a technique used in image steganography where data is concealed within the frequency domain of an image, particularly in the context of JPEG compression. It leverages the mathematical transformation properties of DCT, which converts pixel values into a frequency representation. During DCT embedding, the image is first divided into blocks, typically 8x8 pixels each, and then the DCT is applied to these blocks. The discrete cosine transform identifies the image's frequency components, and the least significant coefficients within these transformed blocks are subtly modified to encode the hidden data. Since DCT coefficients are less perceptually significant, these alterations are less likely to be visually noticeable. However, DCT embedding may have limitations in terms of data capacity compared to other methods like LSB embedding. Still, it offers a higher degree of steganalysis resistance as changes in the frequency domain are less evident to potential attackers. DCT embedding is often chosen when balancing data capacity and steganographic robustness is crucial, especially in the realm of JPEG images. WT(Wavelet Transform) embedding: Wavelet transform embedding is a sophisticated technique in image steganography where data is concealed within the frequency domain using wavelet transforms. This method capitalizes on the wavelet transform's ability to decompose an image into multiple frequency sub-bands, providing a rich landscape for data embedding. During wavelet transform embedding, the image is transformed into different scales and orientations, each represented as a sub-band. The embedding process focuses on altering coefficients within these sub-bands, often targeting less perceptually significant components. This enables the hidden data to be seamlessly integrated into the image while minimizing visual distortion. One of the key advantages of wavelet transform embedding is its robustness against steganalysis, as changes in the frequency domain are typically less conspicuous to potential attackers. Additionally, it can offer a higher data capacity compared to simpler techniques like LSB embedding. Wavelet transform embedding is often chosen when the priority is to maintain both data security and image quality in scenarios such as medical imaging, forensics, or secure communication where preserving the integrity of the visual content is crucial. LSB (Least Significant Bit) embedding: LSB (Least Significant Bit) embedding is a widely-used and straightforward technique in image steganography, employed to conceal data within the least significant bits of the pixel values in a digital image. During LSB embedding, binary data representing confidential information, such as text messages or files, is sequentially inserted into the least significant bit of each pixel's color channel (typically Red, Green, and Blue in color images). Because the least significant bits contribute minimally to the overall pixel value, these subtle alterations are usually imperceptible to the human eye and do not significantly degrade the image quality. LSB embedding offers a balance between data capacity and visual integrity, making it a popular choice for covert communication and data protection. However, it may not be entirely robust against advanced steganalysis techniques, and excessive alterations may lead to image quality degradation. Nevertheless, its simplicity, efficiency, and minimal computational overhead make LSB embedding a practical choice for Android-based steganography, especially when concealing information within images shared through various

communication platforms. SS(Spread Spectrum) embedding: Spread spectrum embedding is an advanced technique in steganography that operates by spreading the hidden data across the entire spectrum of the host signal. Unlike traditional methods like LSB embedding, which focus on altering specific bits of pixel values, spread spectrum steganography introduces imperceptible changes across the entire signal. This approach disperses the hidden data using a carrier signal, making it exceedingly resistant to detection. In spread spectrum embedding, the host signal is combined with a pseudorandom noise sequence, also known as a spreading code, which acts as the carrier for the concealed data. The hidden information is modulated by the spreading code across the signal, effectively distributing it uniformly throughout. This makes it extremely challenging for steganalysts to identify and isolate the concealed data, as it appears as noise rather than structured alterations. Spread spectrum embedding offers remarkable robustness against various steganalysis techniques, ensuring that the hidden information remains secure and undetectable. However, it comes at the cost of reduced data capacity compared to simpler methods like LSB embedding. This technique finds utility in applications where data security and resistance to detection are paramount, such as military communications and highly sensitive data transmission. Hybrid embedding: Hybrid embedding is a versatile and advanced approach in steganography that combines multiple embedding techniques within a single framework to achieve a balance between data capacity, security, and steganalysis resistance. This method capitalizes on the strengths of different embedding methods to enhance the overall performance of data concealment. In hybrid embedding, a steganographic algorithm dynamically selects the most suitable technique for embedding based on the characteristics of the host image and the data to be concealed. For instance, it may employ LSB embedding for regions where subtle changes are imperceptible and utilize frequency domain embedding like DCT or wavelet transform in areas where resistance to steganalysis is crucial. Additionally, it can employ spread spectrum or quantization-based methods for added security. The hybrid approach offers a flexible solution that adapts to the unique requirements of each steganographic task. It maximizes data capacity while minimizing the risk of detection by steganalysts, making it a powerful tool in scenarios where balancing data security, integrity, and steganalysis resistance is paramount, such as secure communication and digital watermarking. However, the complexity of hybrid embedding may lead to higher computational demands. The best module for Android will depend on the specific needs of the application. If security is the primary concern, then a technique like DCT or wavelet transform embedding should be used. If computational power is a constraint, then LSB embedding may be a better choice. And if the application needs to be able to hide large amounts of data, then a hybrid embedding scheme may be the best option. Here are some specific modules that have been proposed for Android: SD(SteganoDroid): As of my last knowledge update in September 2021, there was no specific mention of "SteganoDroid" as a widely recognized or established Android based steganography tool or application. It's possible that new steganography tools or apps have emerged since then, and "SteganoDroid" might be one of them. If "SteganoDroid" is a steganography tool for Android, it would likely function to conceal and extract hidden data within digital images on Android devices, possibly using techniques such as LSB embedding, DCT embedding, or other steganographic methods. Users of such tools typically employ them to protect sensitive information, enable secure communication, or preserve data integrity. However, the specific features, capabilities, and security of "SteganoDroid" would depend on its design and implementation, and it's essential to exercise caution when using steganography tools to ensure ethical and legal compliance. For up-to-date and accurate information about "SteganoDroid," I recommend visiting the official website or app store listing for the tool or referring to user reviews and documentation specific to that application. SteganographyLib: This is a library that provides a number of steganography functions for Android, including embedding, extraction, and verification.

5. RESULT AND DISCUSSION

Android-based image steganography, like any form of steganography, carries significant importance in various domains due to its ability to hide information within images. Here are some of the key significance and applications of Android-based image steganography: Privacy and Confidentiality: Image steganography can be used to hide sensitive information within images, making it a valuable tool for maintaining privacy and confidentiality. For example, individuals and organizations can use it to securely exchange sensitive data or messages without drawing unwanted attention. Secure Communication: Android-based image steganography can enhance the security of communication channels, especially in environments where encrypted messaging apps might be restricted or monitored. It allows users to communicate covertly within the visual data. Data Hiding: This technique can be used for covert data storage and transmission. You can hide data within images, which is useful for scenarios where concealing the existence of the data is crucial. Digital Watermarking: Digital

watermarking is a form of steganography where information is hidden within images for copyright protection and ownership verification. In the Android context, this can be applied to protect intellectual property in images, videos, or audio. Covert Communication: In situations where overt communication is restricted or monitored, Android-based image steganography can provide a means of covert communication. This can be useful for activists, journalists, and individuals in authoritarian regimes. Forensics and Investigations: Image steganalysis, the process of detecting hidden data within images, is a significant aspect of digital forensics. Android-based image steganography tools and techniques can be used by both investigators and those trying to conceal information. In summary, Android-based image steganography has significant implications for privacy, security, authentication, and covert communication. Its applications range from protecting sensitive information to artistic expression and security testing. However, it's important to use such techniques responsibly and ethically, as they can be used for both legitimate and malicious purposes.STRENGTHS: Widespread Platform: Android is one of the most widely used mobile operating systems globally. Developing steganography tools and applications for Android means that they can potentially reach a vast user base. Ease of Use: Android devices typically have user-friendly interfaces, making it easier for individuals to use steganography apps without requiring advanced technical knowledge. Portability: Android devices are portable and widely available, allowing users to carry steganography tools wherever they go. This makes it convenient for secure communication or data hiding on the go. Diverse Hardware: Android devices come in various forms, from smartphones and tablets to wearables and embedded systems. This diversity of hardware provides opportunities for creative and specialized steganography applications. Multimedia Capabilities: Android devices are known for their multimedia capabilities. They can handle images, audio, and video, making it possible to apply steganography to a wide range of media types. Integration with Cameras: Many Android devices are equipped with high-quality cameras. This integration enables users to capture images directly and apply steganography techniques, potentially simplifying the process. App Ecosystem: The Android ecosystem boasts a large number of apps and developers. This means that there are existing steganography apps and libraries available for Android, making it easier for developers to build upon existing work.LIMITATIONS: Image Quality Loss: Many steganography techniques can result in a loss of image quality, especially if the hidden data takes up a significant portion of the image. This can make the steganographic image appear different from the original. Limited Data Capacity: The capacity for hiding data within an image is limited. Depending on the technique and image size, you may not be able to hide large volumes of data. Detection Risks: Sophisticated steganalysis techniques can potentially detect steganographic content. While some methods are resistant to detection, no technique is entirely immune, and advances in steganalysis can pose risks to the security of hidden data. Compatibility Issues: Steganography tools and techniques developed for one Android device or version may not work as intended on all Android devices due to variations in hardware, operating system versions, and manufacturer customizations. Storage Limitations: Android devices have limited storage capacity, and storing numerous steganographic images or files could consume valuable space on the device. Compression and Editing: Image compression and editing, such as resizing or converting to different formats, can potentially disrupt or reveal hidden data, reducing the robustness of the steganography technique. Encryption Challenges: Combining encryption with steganography can be complex. While encryption secures the data, it can also increase the detectability of steganographic content. Security Risks: If used improperly or maliciously, Android-based image steganography can pose security risks. For example, it can be used to hide malware or conduct covert communication for illegal activities.COST-BENEFIT ANALYSIS: Development Costs: Developing a robust Android-based steganography solution can be costly in terms of software development, testing, and ongoing maintenance. Resource Consumption: The process of encoding and decoding steganographic data may require significant computational resources, which can affect device performance. Usability Challenges: Users may need to learn how to use steganography tools properly, and they might require additional software for decoding, potentially increasing the complexity of communication. Security Risks: Misuse or improper implementation of steganography can pose security risks, especially if it falls into the wrong hands. Legal and Ethical Considerations: Compliance with local and international laws and ethical considerations is necessary to avoid legal issues or misuse of steganography. Maintenance and Updates: Keeping steganography tools up to date and secure can require ongoing effort and resources. Compatibility Issues: Ensuring compatibility with various Android devices, versions, and user scenarios may require additional development and testing.

6. REFERENCE

R. Chandramouli, M. Kharrazi, and N. Memon, "Image Steganography and Steganalysis: Concepts and Practice", T. Kalker et al. (Eds.): IWDW 2003, LNCS 2939, Springer-Verlag Berlin Heidelberg, 2004, pp. 35–49.

F. A.P. Petitcolas, R. J. Anderson, "On the Limits of Steganography", IEEE Journal of Selected Areas in Communications, 16(4):474-481, May 98, Special Issue on Copyright & Privacy Protection. ISSN 0733- 8716, pp 474-482.

P. Salee, "Model–based Steganography", In: Proceeding of the 2nd International workshop on digital water marking, Seoul, Korea, October 20-22 2003, LNCS, vol.2939, pp. 254-260.

J. Silman, "Steganography and Steganalysis: An Overview", SANS Institute, 2001.

W. Huaiqing and W. Shouzhong, "Cyber Warfare: Steganography vs. Steganalysis", October 2004, Vol. 47, No. 10 communication of ACM, pp. 76-82.

A. Cheddad, J. Condell, K. Curran, & P. Mc Kevitt, (2010). Digital image steganography: Survey and analysis of current methods. Signal Processing, Vol 90, Issue 3, March 2010, pp. 727-752.

M. Kharrazi, H.T. Sencar and N. Memon, "Cover Selection for Steganographic Embedding", IEEE International Conference on Image processing, 8-11 oct 2006, Atlanta USA, pp. 117-120.

A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems" Information Hiding. 3rd International Workshop, p. 61–76, 1999.S. Dumitrescu, X. Wu, and N. Memon, "On steganalysis of random lsb embedding in continuous-tone images" IEEE International Conference on Image Processing, ROchester, New York., September 2002.

J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color and grayscale images" IEEE Multimedia Special Issue on Security, pp. 22–28, October November 2001.

J. Fridrich, R. Du, and L. Meng, "Steganalysis of lsb encoding in color images," ICME 2000, New York, NY, USA. 39

A. Westfeld, "Detecting low embedding rates" Information Hiding. 5th International Workshop, p. 324–339, 2002.