e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

STEGANOGRAPHY ON ANDRIOD

Prof. P.A. Shinde*1, Uthkarsh Mendhe*2, Vinay Hire*3, Saugandh Deshmukh*4, Janhavi Jagtap*5

*1,2,3,4,5Computer Department, Navsahyadri Group Of Institute, Pune, Maharashatra, India.

https://www.doi.org/10.56726/IRJMETS52723

ABSTRACT

This research investigates the realm of steganography on Android platforms, focusing on techniques to clandestinely embed information within digital media. The study explores various methodologies such as LSB (Least Significant Bit) embedding, frequency domain manipulation, and encryption enhancements for robust data concealment and secure communication. An in-depth analysis of existing Android steganography applications is conducted, evaluating their efficacy in concealing data within images, audio, and video files while maintaining visual and auditory integrity. The study also considers the potential impact of these techniques on data security and privacy in Android environments. By examining the strengths and limitations of different steganographic approaches on Android, this research contributes to a comprehensive understanding of covert communication mechanisms and their applicability in modern mobile platforms, emphasizing the importance of privacy-preserving techniques in digital information exchange.

Keywords: Analysis, Investigation, Research, Steganography, Android.

I. INTRODUCTION

Steganography, an ancient art with roots dating back to ancient Greece and Rome, has seen a resurgence in the digital age. In essence, steganography involves concealing secret information within seemingly innocuous carriers, thus enabling covert communication between parties. With the exponential growth of digital communication channels and the widespread adoption of mobile devices, the integration of steganographic techniques into modern platforms has become increasingly relevant. Among these platforms, Android, the most popular mobile operating system globally, presents a fertile ground for exploring the implementation of steganography. As Android devices have become indispensable tools for communication, entertainment, and productivity, the potential for embedding hidden messages within various media formats on these devices is vast. This integration not only facilitates secure communication between users but also raises pertinent questions regarding data privacy, digital forensics, and cybersecurity. The aim of this research paper is to delve into the realm of steganography on Android, exploring the methodologies, challenges, and implications of concealing information within the Android ecosystem. By examining existing literature, discussing relevant concepts and techniques, and presenting practical implementations, this paper seeks to contribute to the understanding of steganography in the context of modern mobile computing.

II. METHODOLOGY

Method and analysis which is performed in your research work should be written in this section. A simple strategy to follow is to use keywords from your title in first few sentences.

1. Research and Planning

Literature Review: Begin by conducting an extensive review of existing literature on steganography techniques, especially focusing on their application and adaptation for Android platforms. Identify key concepts such as LSB embedding, frequency domain manipulation, and encryption-based steganography.

Requirement Analysis: Define the project's objectives, scope, and constraints. Determine the types of digital media files to be used (e.g., images, audio, video), the level of security required, and the target audience or users for the Android application.

2. Development and Implementation:

Design Phase: Create a detailed design of the Android steganography application, including user interfaces for embedding and extracting hidden data, algorithms for steganographic techniques, and integration of encryption methods (if applicable). Plan the overall architecture and data flow within the application.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

Implementation: Develop the Android application using appropriate tools and technologies such as Java or Kotlin for coding, Android Studio for development, and relevant libraries or APIs for media manipulation and encryption. Implement steganography techniques based on the chosen algorithms, ensuring they are effective in hiding data while maintaining media integrity.

Testing and Evaluation: Conduct thorough testing of the Android steganography application to verify functionality, performance, and security. Test different scenarios such as embedding and extracting data from various media files, testing on different Android devices, and evaluating the robustness of hidden data against detection. Gather feedback from testers and make necessary refinements to the application.

Documentation: Document the entire project, including design decisions, implementation details, testing procedures, challenges faced, and lessons learned. Prepare user manuals or guides for using the steganography application effectively. Generate a comprehensive report summarizing the project methodology, results, and conclusions.

III. MODELING AND ANALYSIS

Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprises of four sub-processes. The first round process is depicted below:

Byte Substitution (Sub Bytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shift rows

Each of the four rows of the matrix is shifted to the left. Any entries that "fall off" are re-inserted on the right side of row. Shift is carried out as follows –

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

Mix Columns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Add round key

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round, then the output is the cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

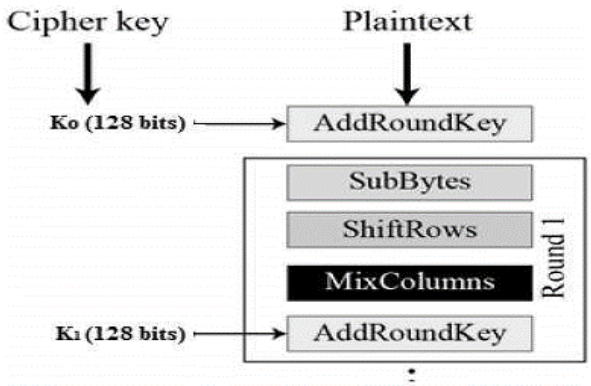

Figure 1: Encryption process

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

LSB (least significant bit)

In computing, the least significant bit (LSB) is the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. The LSB is sometimes referred to as the right-most bit, due to the convention in positional notation of writing less significant digits further to the right. It is analogous to the least significant digit of a decimal integer, which is the digit in the ones (right-most) position.

It is common to assign each bit a position number, ranging from zero to N-1, where N is the number of bits in the binary representation used. Normally, this is simply the exponent for the corresponding bit weight in base-2 (such as in 231...20). Although a few CPU manufacturers assign bit numbers the opposite way (which is not the same as different endianness), the term least significant bit itself remains unambiguous as an alias for the unit bit.

By extension, the least significant bits (plural) are the bits of the number closest to, and including, the LSB.

The least significant bits have the useful property of changing rapidly if the number changes even slightly. For example, if 1 (binary 00000001) is added to 3 (binary 00000011), the result will be 4 (binary 00000100) and three of the least significant bits will change (011 to 100). By contrast, the three most significant bits (MSBs) stay unchanged (000 to 000).

Least significant bits are frequently employed in pseudorandom number generators, steganographic tools, hash functions and checksums.

IV. RESULTS AND DISCUSSION

Original Image

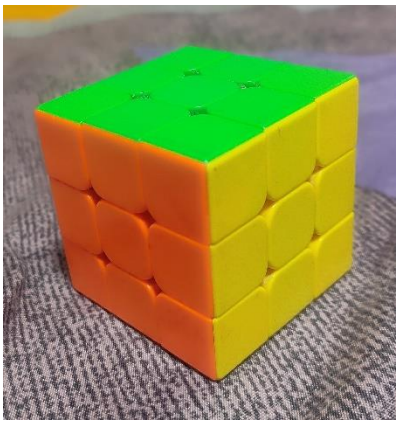

Image with no hidden text (Before steganography)

Stego Image

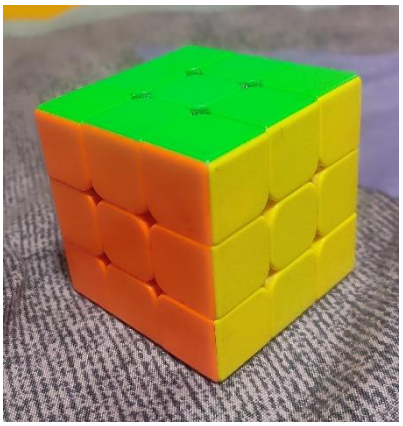

Image with hidden text (After steganography)

As we discussed steganography simply writes data to image by replacing bits or bytes of the image, so when an image having updated text is modified it may happen that the information stored in image is lost.

For example if I crop or resize the image and the bits that contain our information are affected due to this operation then that particular information does not remain accessible.

Similarly other operations like changing image settings or applying new effects may also cause loss of data.

But if this technique is used with internal communication or use then there are rare chances of modifying the image, because users know about the image and the hidden text and will not make any changes.

Especially in smart phone the image with hidden text is mostly communicated in the form of MMS. And the information remains there even same image is sent to many recipients either at a time or from one to another. So there are very rare chances of editing the image in smart phone itself.

V. CONCLUSION

The Least Significant Bit (LSB) method is a basic way to hide secret data inside digital images on Android devices. In this project, we used the LSB technique and tested it on different Android phones and tablets. The LSB approach is easy to implement, but it has some drawbacks. It can only hide a small amount of data without causing visible changes to the image. Additionally, it is not very secure against detection by special analysis tools. While the LSB method works for simple data hiding needs on Android, it may not be the best choice when keeping the data very private and secure is crucial. For those situations, exploring more advanced data hiding

techniques designed specifically for Android would likely provide better protection against unauthorized access to the hidden information. Overall, the LSB technique serves its purpose for basic image steganography on Android, but more robust methods are recommended for applications requiring stronger security guarantees.

VI. REFERENCES

[1] Puteri Awaliatush Shofro, Kiki Widia, Eko Hari Rachmawanto "Improved Message Payload and Security of Image Steganography using 3-3-2 LSB and Dual Encryption" Journal of Theoretical and Applied Information Technology, vol. 95, pp. 1669-1679, April 2018.

[2] Seyed Hesam Odin Hashemi, Mohammad-Hassan Majidi, Saeed Khorashadizadeh "Color Image steganography using Deep convolutional Autoencoders bAESd on ResNet architecture, vol.518, p.052003, 2019.

[3] E. S. I. Harba, "Secure data encryption through a combination of AES, RSA and HMAC," Engineering, Technology & Applied Science Research, vol. 7, pp.1781-1785,2017.

[4] S. Almuhammadi and A. Al-Shaaby, "A survey on recent approaches combining cryptography and steganography," Computer Science Information Technology (CS), pp.63-74,2017.

[5] A. Desoky, Noiseless steganography: The key to covert communications CRC Press, 2012, p. 275.

[6] S. Tanna, Codes, Ciphers, Steganography & Secret Messages, U K:Answers 2000 Limited, 2020.

[7] F. . Q. A. Alyousuf, R. Din and A. J. Qasim, "Analysis review on spatial and transform domain technique in digital steganography," Bulletin of Electrical Engineering and Informatics, vol. 9, pp. 573-581,2.