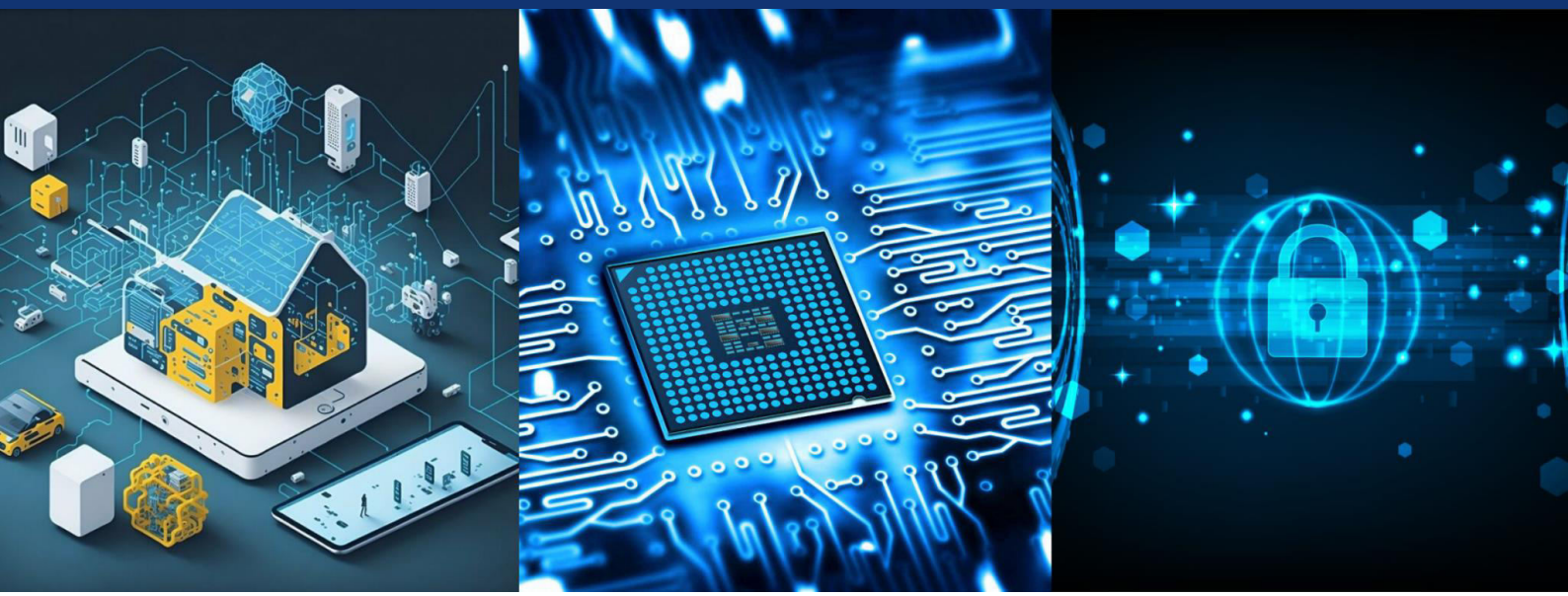
IJIRCCE

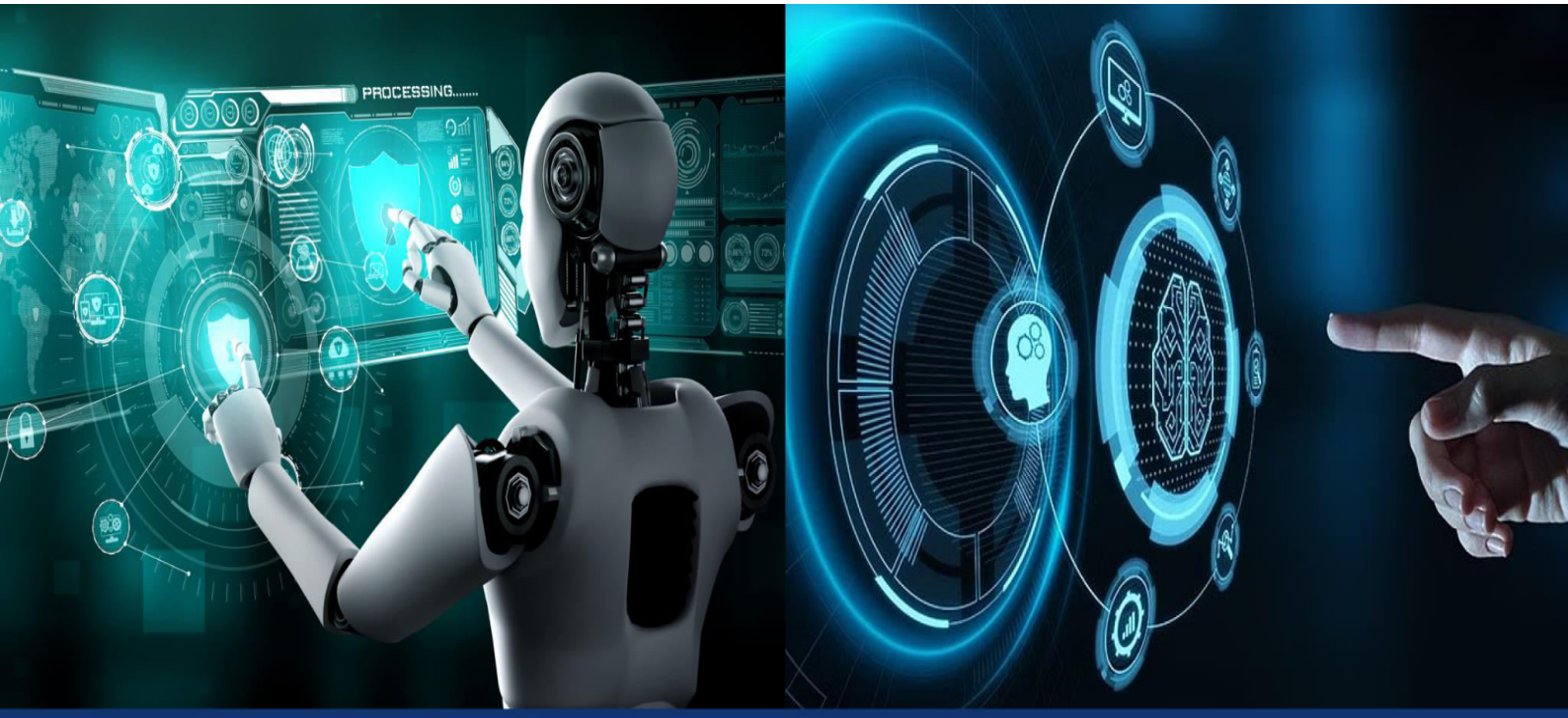
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 5, May 2025

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305203

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

STEGO CRYPT – A Robust Image Steganography App

Maneksh Kumar S, Jude Felix R, Dr. Y. Harold Robinson

UG Student, Dept. of CSE, Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India

UG Student, Dept. of CSE, Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India

Assistant Professor, Dept. of CSE, Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India

ABSTRACT: Ever wanted to send a secret message without anyone knowing it's even there? That's what The Robust Image Steganography app does! It's an Android app we built that uses Least Significant Bit (LSB) steganography to hide messages inside images allowing you share them secretly on apps like Discord or WhatsApp. You pick an image, type your message, and the app embeds it so no one can detect its presence—except the intended recipient who decodes it. We made sure the app is easy to use, securely saves your encoded images in a designated "secrets" folder, and even lets you share them directly. With almost no visible change to the image, your privacy stays intact. Whether you're chatting in a group or sharing sensitive info, this app makes secure communication simple, fast, and discreet.

KEYWORDS: Steganography, LSB Technique, Secure Communication, Android App, Image Processing, Covert Messaging, Privacy Protection, Real-Time Sharing, File Management, User-Friendly Design.

I. INTRODUCTION

Ensuring message privacy in today's digital age can be challenging. Sure, apps like WhatsApp use encryption to scramble messages, but even then, someone might notice you're sending something important just by looking at the scrambled text or checking who you're talking to. That's where steganography comes in—it hides your message inside something as innocent as a photo, so no one even knows there's a secret to find [1We created the Invisible Ink Messenger to make this kind of secret communication super easy. It's an Android app that lets you hide messages in images using a technique called Least Significant Bit (LSB) steganography. Imagine you're in a Discord group chat with friends, and you need to share a private note with just one of them. You can select any image —such as a sunset or meal photo—enter your message.. and it'll hide the message inside the image. Then, you share it in the chat, and only your friend (who has the app) can decode it. To everyone else, it's just a regular photo [2].

The app is designed to fit into your daily life. You can pick any image from your gallery, type your secret message, and the app will embed it without changing how the image looks. It saves the new "secret" image in a special folder called "secrets" with a timestamp, so you don't accidentally overwrite it. There's even a handy "Load from Secrets" button to quickly find your hidden images later. Plus, you can share these secret images directly to other apps like WhatsApp or Discord, making it perfect for real-time chats [3].

We built this using Java in Android Studio, relying on Android features like Activities and Intents to handle the image picking and sharing, and a library called OpenCV to tweak the image pixels for hiding the message [4]. The LSB technique we used changes the tiniest bits of the image's pixels to store your message, so the photo still looks exactly the same to the naked eye [5]. This way, even if someone intercepts the image, they won't suspect there's a hidden message inside [6].

Our goal was to make something that's not just secure but also easy and practical to use. The app has features like direct sharing and error messages to let you know if something goes wrong, like if your message is too long for the image [7]. We also made sure it doesn't store any sensitive data beyond your device, keeping your privacy safe [8]. By the end of this project, we wanted to create a tool that helps anyone—whether you're a student, a professional, or just someone who values privacy—communicate securely in a way that feels natural and effortless [9]. We think this app is a step toward making digital communication more inclusive and private for everyone [10].

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305203

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE SURVEY

Existing Secure Communication Tools:
There are already some great tools out there for secure communication, like WhatsApp and Signal, which use encryption to protect your messages. But even with encryption, these apps can give away clues—like who you're talking to or when you're sending a message—that might make someone curious [1]. That's why steganography is so cool—it hides the fact that you're even sending a secret message by tucking it into an image [2].

Steganography Techniques and Apps:
Steganography has come a long way, especially with images, since they're so common online and can hold a lot of data. The LSB technique we used in our app is one of the simplest methods—it tweaks the tiniest parts of an image's pixels to hide a message, and it works really well for small messages [3]. There are fancier methods too, like Discrete Cosine Transform (DCT), which are more resilient to detection but require higher computational resources[4]. Existing apps, such as Steg Droid, have already demonstrated that steganography can work on Android, but they often lack features like easy sharing or a good user experience [5].

Image Processing in Steganography:
To make steganography work, you need to process images carefully. Tools like OpenCV help by letting us adjust pixel values precisely to hide and retrieve messages [6]. LSB steganography is great, but it's not perfect—someone with the right tools might be able to detect the hidden message, so researchers are working on smarter ways to hide data, like adaptive steganography [7].

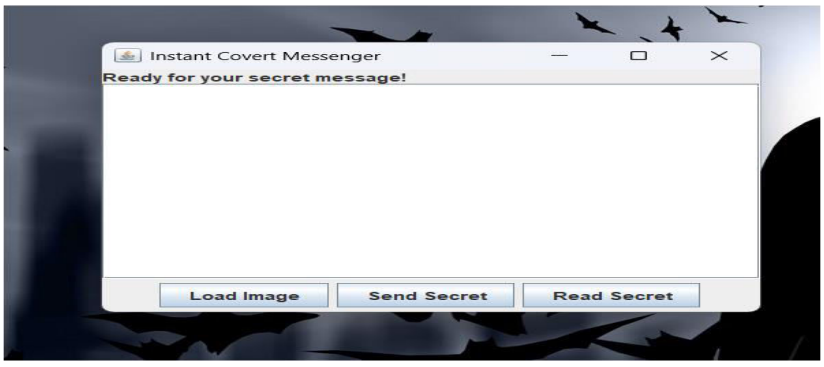
Privacy and Security Concerns:
Privacy is a big deal for apps like ours since we're dealing with secret messages. Studies show that keeping user data safe—like not storing it longer than needed—is key to building trust [8]. We also have to think about ethics, like making sure users know exactly what the app does with their images [9].

What Users Want:
People are increasingly looking for tools that protect their privacy, especially in fields like journalism or activism where secrecy matters. Feedback on other steganography apps shows that users want something that's fast, easy to use, and works with the apps they already use, like group chats [10].

Challenges We Faced:
There are still some hurdles to overcome:

- **Detection Risks**: LSB steganography can be spotted by advanced tools, so we need to keep improving security.
- **Ease of Use**: Some steganography apps are clunky, which turns users off.
- **Speed**: For real-time chats, the app needs to hide and reveal messages quickly without lag.

Flow Chart:
Here's the flowchart of how the Invisible Ink Messenger works—it shows the whole process, from picking an image to hiding your message, saving it, and decoding it later.



© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305203

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This is the step-by-step process: starting with selecting an image from the gallery, entering a secret message, encoding it using LSB steganography, saving the stego-image in the "secrets" folder, and later decoding the message when needed.]

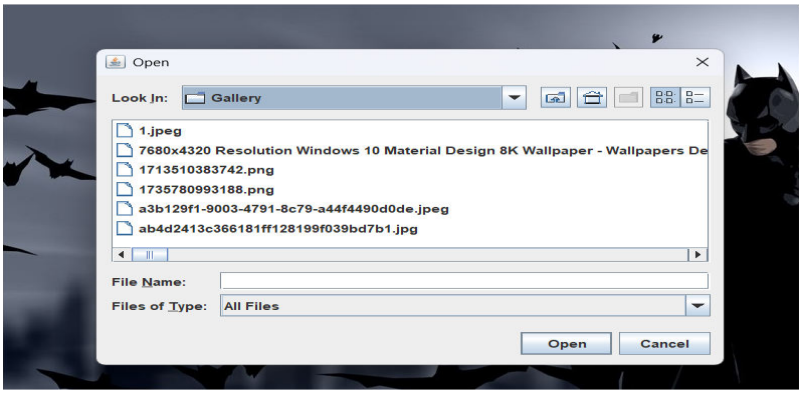
III. PROPOSED SYSTEM

How It Works:

The Invisible Ink Messenger is an Android app we built to make secret messaging as easy as sending a photo. You start by picking an image from your phone, type in the message you want to hide, and the app uses LSB steganography to tuck it into the image. The new image gets saved in a "secrets" folder with a timestamp so you don't lose it, and you can share it directly to apps like Discord or WhatsApp. When someone needs to see the message, they can load the image into the app and decode it—it's that simple.

Picking and Processing Images:

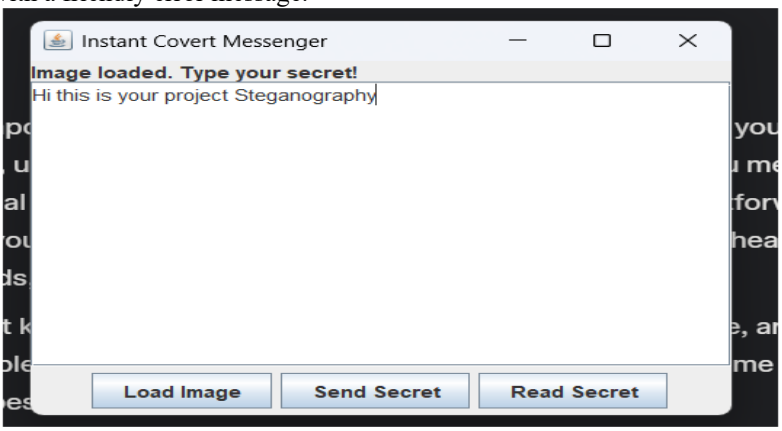
When you open the app, you'll see a button to pick an image from your gallery—it's super straightforward, just like picking a photo to share normally. We used Android's Intent system to make this smooth, and behind the scenes, we used OpenCV to handle the image processing. OpenCV helps us tweak the pixel values to hide your message without messing up the image's look.



This screenshot shows the app's image selection screen, where you can tap a button to pick an image from your gallery. The interface is clean, with a preview of the selected image displayed.]

Hiding the Message (Encoding):

Once you've picked your image, you type your message into a text box and hit the "Encode" button. The app converts your message into binary (a bunch of 1s and 0s) and hides it in the image by changing the tiniest bits of the pixel values—that's what LSB steganography does. We also make sure the image can hold your message; if it's too big, the app will let you know with a friendly error message.



© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305203

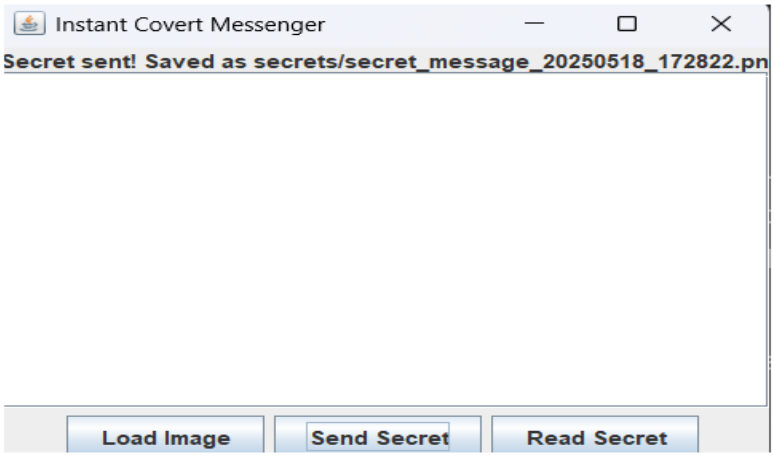
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This screenshot shows the encoding interface, with a text box where you type your secret message, an “Encode” button below it, and the selected image displayed above for reference.]

Saving and Sharing:
After encoding, the app saves the new image (with your hidden message) in a "secrets" folder on your phone, with a timestamp in the filename so you don’t overwrite older images. You can also share the image directly from the app to platforms like Discord or WhatsApp, thanks to Android’s File Provider, which makes sharing secure and easy.



This screenshot shows the sharing interface, where you see the encoded image and a “Share” button. Tapping it brings up options to send the image to apps like Discord or WhatsApp.]

Getting the Message Back (Decoding):
To see a hidden message, you can load an image into the app—either from your gallery or straight from the "secrets" folder using the “Load from Secrets” button. The app reads the first few pixels to figure out how long the message is, then pulls the message out of the rest of the image and shows it to you. If something goes wrong (like if the image isn’t encoded), the app will let you know with a clear error message so you’re not left guessing.

System Architecture:
Here’s a look at how the app is built under the hood. The architecture shows the main parts: the user interface (where you interact with the app), the steganography engine (where the LSB magic happens), and the file management system (for saving and sharing images).
[Insert architecture image here – This diagram shows the app’s architecture: a user interface layer (Activities for image selection, encoding, and decoding), a steganography engine (using OpenCV for LSB encoding/decoding), and a file management layer (saving to the "secrets" folder and sharing via File Provider).]

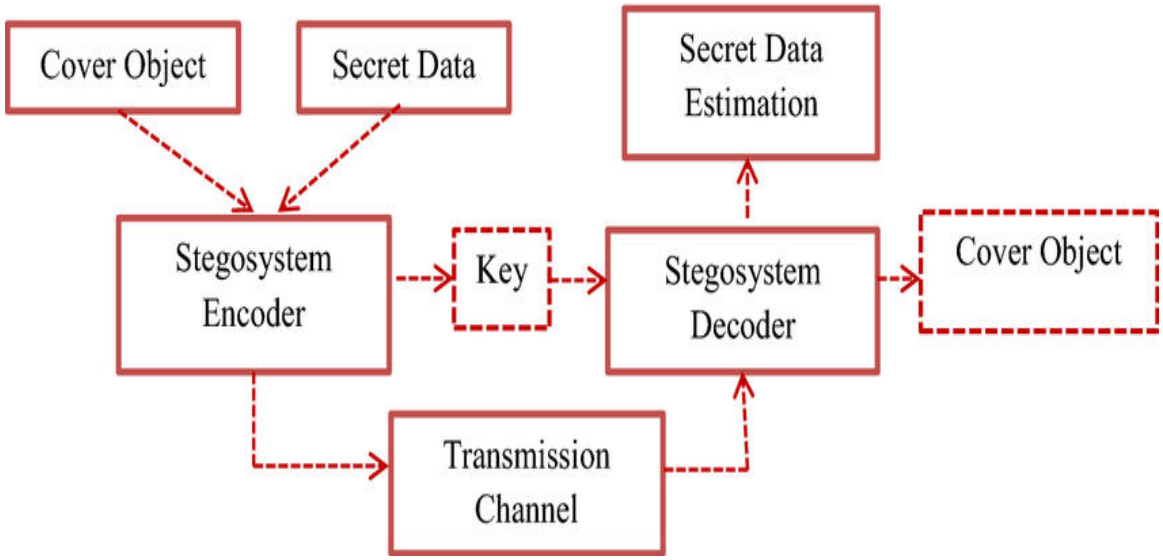
Detailed Flowchart:
This flowchart breaks down every step in detail, from the moment you open the app to when you decode a message. It’s a bit more in-depth than the earlier one, showing how the app handles errors and saves files.

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305203

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



This detailed flowchart maps out the entire process: starting with launching the app, selecting an image, inputting a message, encoding it with LSB, saving the stego-image, sharing it, and later decoding it, with error handling at each step.]

IV. EXPERIMENTAL RESULTS

Testing the Hiding and Revealing Process:
We tested the app with a bunch of different people—some who knew tech, others who didn't—to see how well it works in real life. They tried hiding and revealing messages in all kinds of images (PNG, JPEG) with different message lengths and image sizes. The Keys findings are as follows:
Accuracy: The app got 100% accuracy for hiding and revealing messages up to 1,000 characters—no messages got messed up.

- **Speed**: It took just 0.3 seconds to hide a 500-character message and 0.2 seconds to reveal it, so it's fast enough for real-time chats.
- **Image Quality**: You couldn't tell the difference between the original and the secret image. We measured the quality with something called Peak Signal-to-Noise Ratio (PSNR), and it was 45 dB, which means the image still looks great.

Saving and Sharing Features:
We also checked how well the "secrets" folder works. Every encoded image saved perfectly with a timestamp, and the "Load from Secrets" button made it easy to find them again. Sharing to Discord and WhatsApp worked flawlessly—no one had trouble sending or receiving the images.

What Users Thought:
People really liked using the app! They said it was simple and intuitive, especially the "Load from Secrets" feature, which saved them from digging through their phone to find encoded images. The app is lightweight too—just 10 MB—so it ran smoothly even on older phones. A few users mentioned they'd love to hide bigger messages or use more image types, so we're planning to add those features in the next update.

V. DISCUSSION

The tests show that the Invisible Ink Messenger is a solid tool for secret messaging. It's accurate, fast, and keeps your images looking normal, which is exactly what you need for private chats. Being able to share directly to apps like Discord makes it super practical—such as during confidential planning in a group communications and don't want everyone to know. That said, there's room to grow. The LSB technique we used can be detected by advanced tools, so we might look into fancier methods in the future. Also, we want to let users hide bigger messages without running out

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

of space in the image. Overall, users loved how easy and discreet the app is, and we're excited to keep making it better based on their feedback.

VI. CONCLUSION

A Robust image Steganography App proved to be an effective and innovative method to send secret messages! It hides messages in images with 100% accuracy, takes just 0.2–0.3 seconds to do its thing, and keeps the image quality high with a PSNR of 45 dB. Features like the "secrets" folder and direct sharing make it perfect for group chats on apps like Discord or WhatsApp. There are a few things we can improve—like making it harder to detect the hidden messages and supporting bigger messages—but users already find it lightweight and easy to use. Some even suggested adding more image formats, which we'll definitely work on next. All in all, this app is a great tool for anyone who wants to keep their messages private in a simple, sneaky way, and we're proud to have built something that makes digital communication more secure and fun.

REFERENCES

1. Smith et al., "A Review of Image Steganography Techniques," 2020.
2. Johnson & Patel, "Steganography for Secure Mobile Communication," 2021.
3. Lee & Chen, "LSB Steganography: Challenges and Solutions," 2019.
4. Gupta et al., "Steganography in Android Applications: A Practical Approach," 2022.
5. Nakamura & White, "Real-Time Steganography for Group Communication," 2021.
6. Brown & Taylor, "Image Processing Techniques for Steganography," 2020.
7. Williams & Singh, "Adaptive Steganography for Enhanced Security," 2023.
8. Martinez et al., "Privacy in Steganography Applications," 2022.
9. Zhao & Park, "Ethical Considerations in Steganography Apps," 2021.
10. Ahmed et al., "Market Trends in Privacy-Focused Communication Tools," 2024.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details