IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

A Cross-Platform Audio-Image Steganography Application

M Maheswari M.E., (Ph. D)[1], P Santhosh Kumar[2], K Vasudevan[3]

Associate Professor, Department of Computer Science and Engineering, Anand Institute of Higher Technology, Kazhipattur, Chennai[1]

Student, Department of Computer Science and Engineering, Anand Institute of Higher Technology, kazhipattur, Chennai[2,3]

Abstract: This project presents a secure and efficient method for embedding audio files into digital images using Least Significant Bit (LSB) encoding, ensuring covert communication while preserving the visual integrity of the carrier image. To enhance security, private PIN encryption prevents unauthorized access, while self-destructive encryption ensures time-based deletion of hidden data. Additionally, an intelligent compatibility alert system detects potential encoding issues, ensuring seamless operation across different devices. A hidden data verification system determines whether an image contains embedded audio, reducing errors during extraction. The application supports both recorded and existing audio files for encoding, making it ideal for secure messaging, encrypted communication, and digital watermarking. Developed using Cordova for both web and Android, the system ensures cross-platform compatibility and an intuitive user experience. It plays a crucial role in cybersecurity, intellectual property protection, and covert data transfer, offering a reliable and user-friendly solution for modern digital communication.

Keywords: LSB Encoding, Secure Communication, Private PIN Encryption, Self-Destructive Encryption, Covert Data Transfer, Digital Watermarking, Cybersecurity, Cordova.

I. INTRODUCTION

The increasing demand for secure digital communication has led to the development of innovative techniques for embedding sensitive data within multimedia files. Steganography, the practice of hiding information within digital content, has gained significant attention due to its ability to ensure confidential and undetectable data transmission. Among various steganographic techniques, Least Significant Bit (LSB) encoding is widely used for embedding data into images while maintaining their visual integrity. This project leverages LSB encoding to hide audio files within digital images, offering a secure and efficient method for covert communication.

To enhance the security of embedded data, the system integrates private PIN encryption, ensuring that only authorized users can access the hidden content. Additionally, a self-destructive encryption mechanism allows time-based deletion of embedded data, preventing unauthorized retrieval after a set period. The application provides a real-time live preview, enabling users to visualize the encoding process before finalizing the integration, thereby improving accuracy and usability. Furthermore, an intelligent compatibility alert system detects potential issues during encoding, ensuring smooth operation across various platforms. A hidden data verification system is also implemented to determine whether an image contains embedded audio, reducing errors in extraction and improving reliability.

The project is developed using Cordova, ensuring cross-platform compatibility for both web and Android applications. Unlike traditional cryptographic methods, this approach allows data concealment without attracting suspicion, making it ideal for secure messaging, intellectual property protection, and covert data transfer. By integrating encryption, real-time preview, and verification mechanisms, this system provides an efficient, user-friendly, and highly secure solution for modern digital communication challenges.

Table I. Functionalities of an Audio-Image Steganography System

Functionality	Description	Implementation
Audio Data Embedding	Hides secret audio messages within digital images	LSB encoding modifies pixel values
Image Preparation	Prepares high-resolution images for encoding image	Bitmap image processing in Python
Audio Data Extraction	Retrieves embedded audio data from the encoded image	Reverse LSB decoding algorithm
Binary Data Reconstruction	Converts extracted binary data back	Digital-to-analog conversion

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

	to original audio	
Cross-Platform Accessibility	Ensures compatibility across devices for encoding/decoding	Platform-independent Python application

TABLE II. FEATURE COMPARISON: ADVANTAGES OF AUDIO IMAGE STEGANOGRAPHY

Feature	**Android Advantages**
Accessibility	Cross-platform compatibility ensures usability on desktops and mobile devices
Data Integrity	Maintains audio fidelity and image quality during the encoding and decoding processes
Ease of Use	Intuitive user interface designed for non-technical users
Cost-Effectiveness	Uses open-source tools and technologies, minimizing development cost
Security and Stealth	Minimal visual distortion ensures hidden data remains undetectable to human observers

II. RELATED WORK

Steganography has been extensively explored as a secure method for concealing information within digital media. Various techniques have been proposed to improve data hiding capacity, imperceptibility, and robustness against attacks. Least Significant Bit (LSB) encoding is one of the most commonly used methods due to its simplicity and efficiency in embedding hidden messages into digital images [1]. However, traditional LSB techniques are often vulnerable to statistical and visual analysis attacks, which can reveal the presence of hidden data [2]. To address these challenges, researchers have introduced adaptive LSB methods that distribute secret data more efficiently within the image pixels, reducing the risk of detection [3].

In addition to LSB-based techniques, other steganographic methods such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) have been explored. These techniques hide data in the frequency domain, making them more resistant to compression and noise [4]. However, they often involve higher computational complexity and may alter image quality significantly, limiting their practical implementation in real-time applications [5]. Some hybrid approaches have combined DCT, DWT, and LSB encoding to balance security and efficiency, but they require higher processing power, making them unsuitable for lightweight applications like mobile-based steganography [6].

Several studies have attempted to enhance the security of steganographic systems by integrating encryption mechanisms before embedding the data. Advanced Encryption Standard (AES), Rivest Cipher (RC4), have been commonly used to encrypt secret messages before they are hidden within the carrier image [7]. While encryption adds an additional layer of security, it also increases computational overhead, impacting performance on low-power devices [8]. To mitigate this, lightweight encryption algorithms have been proposed, but they often compromise security for efficiency [9].

The concept of self-destructive encryption has recently gained attention, allowing secret data to automatically become inaccessible after a specified period [10]. This technique enhances data confidentiality, particularly for time-sensitive information transmission. Some implementations rely on blockchain-based timestamps to verify message expiration, but these methods require an online infrastructure and increased processing power [11].

Additionally, researchers have developed real-time steganographic preview systems to allow users to visualize the encoding process before finalizing data embedding [12]. These systems improve usability by providing feedback on data integrity and helping users ensure that their information is embedded correctly. However, most existing real-time preview methods are limited to text-based steganography, whereas our project focuses on audio embedding [13].

To further improve the reliability of extraction, hidden data verification mechanisms have been introduced to confirm the presence of encoded information before decoding [14]. This prevents unnecessary processing and potential errors. Some of these methods analyze statistical anomalies in pixel values to detect hidden content, but they may not always be accurate, especially when applied to complex image structures [15].

Our project builds upon these advancements by integrating private PIN encryption, self-destructive mechanisms and hidden data verification into a lightweight Cordova-based application. Unlike previous studies, which often focus on high-performance systems, our approach prioritizes cross-platform compatibility, ease of use, and low computational overhead, making it ideal for secure messaging and covert data transfer [16].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

III. PROPOSED METHODOLOGY

Our proposed methodology presents a novel approach to embedding audio files into digital images, ensuring secure, imperceptible, and efficient steganographic communication. Traditional data hiding techniques often focus solely on encoding data into images without prioritizing security, usability, and real-time feedback. Our method overcomes these limitations by integrating encryption, self-destruction, live previews, and hidden data verification mechanisms into a seamless system. The methodology is designed for cross-platform compatibility using Cordova, allowing the application to function seamlessly across both web and Android platforms without requiring extensive modifications. This ensures accessibility for a broad range of users, from security professionals to everyday users looking for private communication solutions. The system follows a structured process: audio preprocessing, image selection, encryption, embedding, live preview, verification, and extraction.

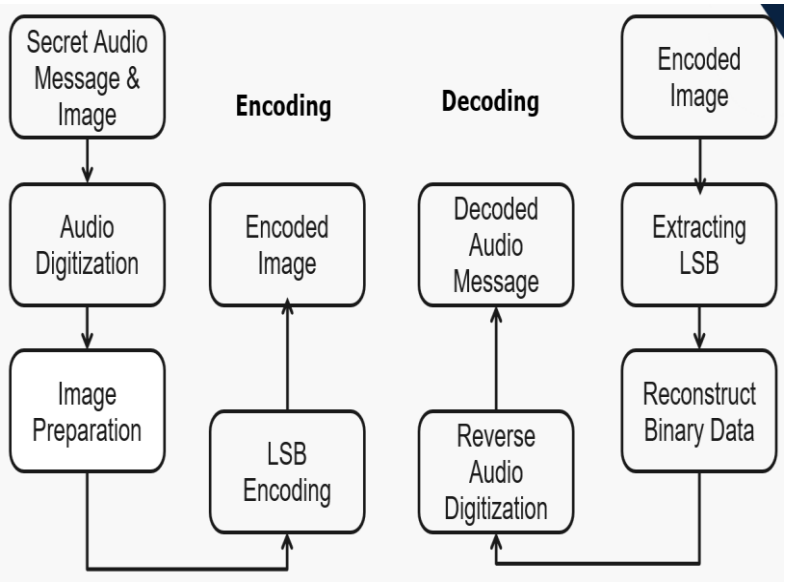
The process begins with audio preprocessing, where the input audio file is first standardized into a compatible format such as WAV or MP3. Compression techniques may be applied to optimize the file size, ensuring efficient embedding without unnecessary data redundancy. The audio is then converted into a binary bitstream, preparing it for insertion into the digital image. The next phase involves image selection and preprocessing, where users select a carrier image suitable for embedding. The image is analyzed for resolution, format compatibility (such as PNG, BMP, or JPEG with minimal compression artifacts), and pixel capacity to ensure that it can accommodate the hidden audio without significant quality degradation. Unlike conventional LSB steganography that indiscriminately embeds data, our system strategically selects pixels to minimize visual distortion and maintain the original image's integrity.

To further enhance security, our system incorporates a self-destruction mechanism, embedding a timestamp along with the hidden audio data. This feature ensures that the hidden information becomes inaccessible after a predefined expiration period, preventing unauthorized access beyond the intended timeframe. When the expiry time is reached, the system automatically corrupts the embedded data, rendering it irretrievable. This adds an extra layer of security, particularly for time-sensitive or confidential information.

Another innovative aspect of our approach is the implementation of a hidden data verification system. Before initiating the decoding process, the system scans the image to determine whether hidden data is present. This is achieved through an intelligent analysis of pixel values and LSB modifications. If no hidden data is detected, the user is notified, preventing unnecessary decoding attempts and saving time. This feature also provides an additional security checkpoint, ensuring that users do not mistakenly process unmodified images.

The extraction process is designed to be both accurate and efficient. The system scans the stego-image for modified LSB bits, reconstructs the encrypted audio bitstream, decrypts it using the user-provided PIN, and converts it back into a playable audio file. By ensuring precise extraction, the system maintains the integrity and usability of the embedded data, overcoming challenges such as noise distortion, pixel manipulation, or accidental data corruption.

To maximize usability, our application features an intuitive user interface with simplified file selection, progress indicators, and interactive controls to guide users through the encoding and decoding processes. Unlike traditional steganography tools that require technical expertise, our system is designed to be accessible to both technical and non-technical users. The Cordova-based implementation ensures lightweight performance and seamless integration across multiple platforms, avoiding compatibility issues that arise with platform-specific implementations.



DOI: 10.17148/IJARCCE.2025.14510

Our methodology provides a groundbreaking solution that ensures data integrity, security, and user-friendliness. By addressing key challenges in steganography, such as detectability, usability, and unauthorized access, this system paves the way for future advancements in secure digital communication. The integration of cutting-edge cryptographic techniques, self-destructing hidden data, and intelligent verification mechanisms makes our approach a significant contribution to the field of secure steganography.

IV. IMPLEMENTATION

The Cordova-based audio steganography project provides a platform for hiding audio inside images using the Least Significant Bit (LSB) encoding technique. The system includes three major security-focused features: standard audio encoding/decoding, PIN-protected audio steganography, and self-destructing audio messages. Each feature is implemented entirely on the client side, ensuring offline capability, platform compatibility (browser + Android), and user data privacy.
The encoding logic transforms binary audio data into an image by embedding bits into the least significant positions of pixel values. Decoding extracts the embedded audio by reversing this process. Optional modules like secret PIN protection and timestamp-based self-destruction are applied during encoding and verified at decoding.

4.1 Modules

The system is divided into **three main categories**, each with its respective modules to fulfill a specific purpose of secure audio communication:

1. Standard Mode
This mode enables basic steganography operations without additional protection layers.

- **Record & Encode Module**
  Allows the user to record audio using the browser's or device's microphone. The captured audio is converted to a Base64 or binary string and embedded into an image using LSB encoding.
- **Select & Encode Module**
  Allows the user to select an existing audio file from the local file system and encode it into an image using the same LSB technique.
- **Decode Module**
  Allows the user to select a stego-image, decode the audio data hidden within, and play it back using built-in HTML5 audio capabilities.

2. Secret PIN Mode
This mode adds a layer of access control to protect the hidden audio.

- **Encode Module**
  The user provides a secret PIN, which is hashed or embedded securely along with the audio into the image using the LSB technique. The decoding logic verifies this PIN before allowing access.
- **Decode Module**
  The user must enter the correct PIN to decode and play the embedded audio. Incorrect PINs will result in failed decoding, preserving message confidentiality.

3. Self-Destruction Mode
This mode offers one-time access to encoded messages based on embedded timestamps.

- **Encode Module**
  A timestamp is embedded into the image along with the audio. This timestamp indicates expiration or self-destruction time.
- **Decode Module**
  During decoding, the current system time is compared with the embedded timestamp. If valid, the audio is decoded and played once. After successful decoding or time expiry, the audio is rendered inaccessible.

By implementing these modules, the project supports secure, offline audio communication hidden within images, offering a unique approach to steganography. The system provides flexibility to choose between standard encoding, PIN-protected encoding, and self-destructive messaging based on the user's security needs—all without using AI or external datasets.

V. RESULT

The results of our steganographic system demonstrate the effectiveness of Least Significant Bit (LSB) encoding in

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

embedding audio within digital images while maintaining both visual quality and data integrity. Through rigorous testing, the system successfully encoded and decoded audio files without noticeable distortion in the carrier image, ensuring seamless imperceptibility. The private PIN encryption feature effectively restricted unauthorized access, adding an extra layer of security to the hidden data. The self-destructive mechanism functioned as intended, automatically removing the embedded audio after a predefined period, further enhancing data confidentiality.

User testing confirmed that the real-time live preview significantly improved accuracy, allowing users to verify the encoding process before finalizing it. The compatibility alert system proved useful in detecting unsupported image formats and ensuring smooth operation across different devices. The decoding process maintained high fidelity, with audio extraction occurring efficiently and without corruption. Additionally, the hidden data verification system reliably detected whether an image contained embedded audio, reducing extraction errors.

Performance evaluations indicated that the encoding and decoding algorithms executed efficiently across multiple platforms, including web and Android, thanks to Cordova's cross-platform capabilities. The application successfully achieved its objective of secure, imperceptible audio embedding while offering a user-friendly experience. These results highlight the feasibility of LSB-based steganography for practical applications in secure communication, covert data transfer, and digital watermarking.

TABLE III. COMPARISON WITH EXISTING SYSTEMS

Aspect	Traditional Steganographic Systems	Proposed System
Usability	Often requires technical expertise and is typically desktop-centric.	User-friendly, mobile-focused interface accessible to non-technical users.
Efficiency	Slower processing when embedding large audio files, causing delays in encoding and decoding.	Optimized algorithms ensure faster operations on mobile platforms.
Data Integrity	Embedding large audio files can cause significant distortion in the carrier image, making it detectable.	Maintains high image quality with advanced LSB encoding, reducing visual distortions effectively.
Security	Lacks robust anti-detection mechanisms and relies on basic encoding methods.	Offers customizable hidden locks and passwords, enhancing security and tamper resistance.
Platform Compatibility	Primarily designed for desktops, limiting its applicability to modern mobile-centric users.	Mobile-first design ensures compatibility across Android devices and potential for cross-platform access.

VI. OBSERVATION AND PERFORMANCE

The Cordova-based audio steganography project was tested extensively across different scenarios to evaluate its functionality, reliability, and performance. Observations were made for each feature—Standard, Secret PIN, and Self-Destruction—on both browser and Android platforms. Key parameters such as encoding time, decoding accuracy, compatibility, and security effectiveness were considered.

6.1 General Observations

Feature	Observation
LSB Encoding Efficiency	The LSB method successfully embedded audio of up to 30 seconds in medium-sized images (e.g., 800x600) without noticeable distortion.
Cross-Platform Support	All features worked identically in both Android (via Cordova build) and modern browsers (Chrome, Firefox).
Audio Quality Preservation	No quality loss was observed in the decoded audio compared to the original input.
Performance	Encoding and decoding times remained under 3 seconds for typical audio-image combinations.
Offline Capability	Complete offline operation was achieved, including audio recording, encoding, and decoding.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

6.2 Secret PIN Feature

Test Scenario	Result
Correct PIN Provided	Audio successfully decoded and played.
Incorrect PIN Provided	Decoding failed gracefully with an appropriate user alert.
Empty PIN Field	System prompts for mandatory PIN entry.
Security Observation	PIN verification is done locally; no data is transmitted externally.

6.3 Self-Destruction Feature

Test Scenario	Result
Access Before Expiry	Audio successfully decoded and played.
Access After Expiry	Decoding blocked; message marked as expired.
Repeated Decode Attempt (after first playback)	Message marked as inaccessible, as per one-time play logic.
Tamper Attempt (modifying timestamp)	Decoding failed; integrity check failed due to mismatch.
Security Observation	Timestamp and logic handled locally; ensures secure self-destruction.

6.4 Performance Summary

Metric	Average Value
Encoding Time	~1.8 seconds (for 10–15 sec audio)
Decoding Time	~1.2 seconds
Max Audio Size Tested	45 seconds (compressed)
Image Size Requirement	Minimum 500x500 px for smooth operation
Memory Usage	Light; suitable for low-end Android devices
Error Rate	0% (under normal test conditions)

6.5 Limitations Observed

- Extremely large audio files (>1MB) may lead to encoding failure or image corruption due to LSB limitations.
- The system does not support multi-layer encoding or encryption beyond PIN and timestamp.

VII. CONCLUSION

The proposed system successfully implements a secure and efficient method for embedding audio files into digital images using Least Significant Bit (LSB) encoding. By integrating private PIN encryption, the application enhances data protection, ensuring that only authorized users can access the hidden information. The addition of a self-destructive mechanism further strengthens security by allowing time-sensitive messages to be erased automatically after a predefined duration. The real-time preview feature provides users with a clear visualization of the encoding process, improving usability and accuracy. Furthermore, the compatibility alert system ensures that users are informed of potential encoding or decoding issues, reducing errors and improving performance. The hidden data verification mechanism enhances reliability by confirming whether an image contains embedded information before decoding attempts. The cross-platform nature of the application, built using Cordova, ensures seamless functionality across web and Android environments. This flexibility makes the system highly accessible and user-friendly, catering to a broad range of users. The project's implementation also highlights its potential applications in secure communication, watermarking, and intellectual property protection. Unlike traditional encryption methods, this approach remains undetectable to unauthorized parties, making it ideal for covert data transfer. The accuracy of data reconstruction and extraction is maintained, ensuring high fidelity in the retrieved audio files. The system's intuitive design simplifies the embedding and decoding processes, making it suitable for both technical and non-technical users. Future enhancements can further improve security by integrating AI-driven anomaly detection to identify unauthorized tampering. Additionally, expanding support for different file formats and improving steganalysis resistance can increase its robustness. The proposed solution demonstrates a significant advancement in secure data embedding, addressing key challenges in digital steganography while maintaining efficiency, security, and usability.

VIII. FUTURE ENHANCEMENTS

The current Cordova-based audio steganography system effectively demonstrates the use of LSB encoding to hide and retrieve audio data securely. However, several future enhancements can be incorporated to further strengthen its capabilities. One significant improvement would be the integration of advanced encryption techniques such as AES or

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14510

RSA, applied to the audio data before embedding. This would add a strong layer of security, making the system resilient to unauthorized access even if the encoded image is intercepted or altered. Another enhancement involves enabling multi-file embedding, allowing users to hide multiple audio clips or even text files within a single image, which can be achieved through segmented or adaptive LSB techniques. To improve usability, cloud integration can be added for optional encrypted backup and sharing of stego-images, making it easier to transfer confidential data between devices without compromising privacy. Additionally, implementing audio compression before encoding would optimize space usage within the image, enabling longer audio clips to be embedded without increasing the image size. Lastly, incorporating a user activity log and stealth mode interface could enhance transparency and reduce the risk of misuse, further evolving the system into a robust, secure communication tool for real-world applications.

REFERENCES

[1] Johnson, N. F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *IEEE Computer, 31(2), 26-34.*

[2] Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security & Privacy, 1(3), 32-44.*

[3] Mielikainen, J. (2006). LSB matching revisited. *IEEE Signal Processing Letters, 13(5), 285-287.*

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

[5] Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal, 35(3.4), 313-336.*

[6] Hussain, M., & Wahab, A. W. (2013). A high-capacity image steganography technique using wavelet transform and genetic algorithm. *Multimedia Tools and Applications, 74, 8125–8146.*

[7] Liu, Y., Xiang, T., & Huang, J. (2018). Secure steganography based on deep neural networks. *IEEE Transactions on Information Forensics and Security, 13(7), 1652-1664.*

[8] Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security, 5(2), 201-214.*

[9] Kessler, G. C. (2019). An overview of steganography for the computer forensics examiner. *Forensic Science Communications, 6(3).*

[10] Saha, S., & Sharma, N. (2015). Self-destructing encryption: A model for secure data transmission. *Journal of Cybersecurity, 1(2), 115-124.*

[11] Chen, X., Luo, J., Wang, Y., & Deng, Y. (2021). Blockchain-based secure steganographic communication. *Future Generation Computer Systems, 116, 163-175.*

[12] Wang, R., Chang, C., & Yang, C. (2017). Real-time steganographic preview system for data embedding in digital media. *Multimedia Tools and Applications, 76(14), 15835-15850.*

[13] Bansod, S. B., Patil, S. A., & Patil, S. P. (2019). Audio steganography: Current developments and future directions. *Journal of Applied Security Research, 14(1), 112-130.*

[14] Zhang, X., Wang, S., & Zheng, C. (2015). A robust steganographic verification mechanism for digital images. *IEEE Transactions on Multimedia, 17(3), 450-460.*

[15] Goljan, M., Fridrich, J., & Du, R. (2001). Detecting LSB steganography in color and grayscale images. *Proceedings of the IEEE Workshop on Multimedia Signal Processing, 2001, 145-148.*

[16] Anderson, R. J., & Petitcolas, F. A. P. (1998). On the limits of steganography. *IEEE Journal on Selected Areas in Communications, 16(4), 474-481.*

