DRM PROTECTION AND FILE ORIGINATIONVERIFICATION

JEGATHA R
Department of Information TechnologySri Sai Ram Institute of Technology Chennai, India
jegatha.it@sairamit.edu.in

JOSEPHINE RUTH FENITHA
Department of Information TechnologySri Sai Ram Institute of Technology Chennai, India
josephine.it@sairamit.edu.in

BHARATH V
Department of Information TechnologySri Sai Ram Institute of Technology Chennai, India
sit20it059@sairamtap.edu.in

AKASH J
Department of Information TechnologySri Sai Ram Institute of Technology Chennai, India
sit20it074@sairamtap.edu.in

AAKASH K
Department of Information TechnologySri Sai Ram Institute of Technology Chennai, India
sit20it003@sairamtap.edu.in

***Abstract*— The increased popularity of Over-The-Top (OTT) content consumption has put content creators and distributors under constant threat from piracy. In order to improve content protection and reduce the danger of piracy, we describe a novel method in this work that uses bit manipulation techniques to embed data within users' streaming video content. Our solution uses Python for effective implementation and interfaces effortlessly into current OTT platforms. Through carefully planned manipulation of the least significant portions of video frames, we are able to secretly embed data, which ensures little disruption to the viewing experience while greatly enhancing content security. Our method offers numerous benefits. First off, by hiding embedded data in the video stream, it provides strong security against piracy by rendering it nearly invisible to unauthorized viewers. This makes sure that the encoded data can be linked to the original subscriber even in cases where the video is illegally distributed, which helps with legal action and piracy prevention. Furthermore, by protecting subscriber credentials and personal data inside the video stream, our technology improves user privacy. In addition to stopping illegal access, this also guarantees the privacy of users' data, reducing worries about data security and privacy. When it comes to market availability, our solution is very affordable and easily integrated into current OTT platforms with little need for infrastructure modifications. Because it is compatible with modern programming languages, it can be easily deployed and scaled, which makes it a desirable choice for movie creators looking to improve content protection without having to make a large financial commitment.**

Keywords—Movie piracy, OTT platforms, Subscriber information, Encode, Secure playback, Unauthorized access, Unique encoding, Content protection, Financial projections.

I. INTRODUCTION

The emergence of online streaming services has changed the entertainment scene recently by giving consumers easy access to a huge collection of films and TV series. But this ease of use also carries the risk of movie piracy, which jeopardizes the integrity of the film industry as a whole in addition to undermining content providers' income sources. Movie sharing and illegal distribution have proliferated, causing significant financial losses and copyright violations.

Our study intends to present a novel solution to safeguard films released on content streaming platforms in order to address this ongoing issue. Our technique is different from typical content protection approaches, which usually only use DRM (Digital Rights Management) and encryption. Instead, we encode subscriber information into the movie files themselves. Through the use of subscriber-provided data such as unique identifiers or user-specific keys we are able to seamlessly integrate this data into the video files so that viewers are unaware of it. There are various benefits associated with this subscriber information-based encoding. First of all, it serves as a system for individualized identification, making it possible to track down individual subscribers and their movie files. Thanks to the ability to directly trace any illegally transmitted copies to the person responsible, efficient tracking and enforcement against unauthorized distribution are made possible.

And by confirming the legitimacy of the individual trying to access the video, the system guarantees safe playback. The system verifies the subscriber's identity using the encoded data in the movie file before allowing playback permissions. Even if unauthorized users are able to access the file, this procedure stops them from accessing the content. Additionally, our project uses privacy-preserving strategies in order to prioritize data privacy. Within the system, subscriber data is handled and kept securely, preserving sensitive user information and complying with applicable privacy laws. The goal of the encoding procedure is to protect subscribers' privacy while also permitting efficient identification and verification.

Our goal in putting this creative strategy into practice is to give OTT platforms a complete solution to safeguard their priceless content. In addition to discouraging possible pirates, our solution gives reputable subscribers more security and confidence. Consequently, this improves the general user experience and increases the potential revenue stream for digital platform providers and content creators.

In the current digital environment, safeguarding films released on digital platforms is essential. By incorporating subscriber information into movie files, our project tackles the problems caused by illegal access and movie piracy. We provide a strong solution to protect the rights of content creators, the income streams of online streaming platforms, and the fun and safe movie-watching experience of authorized users by fusing personalized identification, secure playback, data privacy, and cutting-edge content protection measures.

II. LITERATURE SURVEY

Movie piracy on OTT platforms has become a significant concern for movie studios and OTT providers due to the increasing prevalence of unauthorized access to digital content. To combat this issue, researchers and industry professionals have explored various techniques and strategies for protecting movie content and ensuring secure playback for authorized subscribers. In this literature survey, we review existing research and industry practices related to movie piracy prevention, subscriber authentication, content encryption, and personalized content protection on OTT platforms.

A. Movie Piracy Prevention Techniques

DRM: DRM technologies are commonly used to protect digital content from unauthorized copying and distribution. Researchers have explored the effectiveness of DRM systems in preventing movie piracy on OTT platforms.

Watermarking: Watermarking techniques have been proposed to embed invisible markers into movie content, allowing for the identification and tracking of pirated copies.

Anti-Piracy Technologies: Various anti-piracy technologies, such as content fingerprinting and real-time monitoring systems, have been developed to detect and combat movie piracy on OTT platforms.

B. Subscriber Authentication and Access Control

User Authentication Protocols: OTT platforms employ user authentication protocols, including username/password authentication and token-based authentication, to verify the identity of subscribers and grant access to authorized content.

Multi-Factor Authentication (MFA): MFA methods, such as biometric authentication and SMS-based verification, enhance the security of subscriber accounts and reduce the risk of unauthorized access.

C. Content Encryption and Secure Playback

Encryption Algorithms: Advanced encryption algorithms, such as AES and RSA, are used to encrypt movie content and ensure secure transmission and storage on OTT platforms.

Content Protection Systems: Content protection systems like PlayReady and Widevine provide robust security mechanisms for protecting digital content from unauthorized access and piracy.

Secure Playback Technologies: Secure playback technologies, such as hardware-based DRM solutions and secure media pipelines, enable secure decoding and rendering of encrypted movie content on user devices.

D. Personalized Content Protection

Subscriber-Specific Encoding: Researchers have proposed subscriber-specific encoding techniques that leverage subscriber information to generate unique content encodings, thereby enhancing content security and preventing unauthorized access.

E. System Architecture and Scalability

Scalable Encryption Frameworks: Scalable encryption frameworks and cloud-based security solutions provide scalable and flexible architectures for implementing anti-piracy measures and secure content delivery systems on OTT platforms.

Distributed Content Delivery Networks (CDNs): CDNs play a crucial role in distributing encrypted movie content to users while ensuring high availability, low latency, and scalability.

F. Financial Projections and Business Impact

Revenue Impact Analysis: Industry reports and financial analyses have demonstrated the potential revenue impact of implementing anti-piracy solutions on OTT platforms, including increased revenue streams, subscriber retention, and market competitiveness.

The literature survey highlights the diverse range of techniques and strategies employed to address movie piracy on OTT platforms, including DRM, watermarking, subscriber authentication, content encryption, and personalized content protection. By integrating these approaches into a comprehensive anti-piracy solution, movie studios and OTT providers can effectively protect their content from piracy and ensure a secure and seamless viewing experience for authorized subscribers.

III. ARCHITECTURE

A. Problem Definition

The way we consume digital content has changed dramatically with the rise of OTT platforms, which provide a huge selection of films, TV series, and other media that are available anywhere, at any time. But there's a big problem with this content, that is widespread streaming content piracy.

OTT platforms have seen a sharp increase in popularity, drawing in millions of users globally. The accessibility of digital content has, however, also increased its susceptibility to piracy, endangering platform providers and content creators through unlawful streaming and unauthorized distribution.

Creating a strong and practical approach to stop piracy and improve content security on OTT platforms while protecting user privacy and data security is the main problem. The following issues need to be addressed by this solution: Protect the rights of content creators and platform providers and to stop illegal access, distribution, and piracy of streaming content. Make sure that subscriber data—including personal information and viewing history—is kept private and protected from misuse or unauthorised access. To Investigate encryption methods to protect user data and streaming media, reducing the possibility of interception or manipulation while in transit and finally to create a system that can expand with the user base and change with the dangers faced by OTT providers.

B. Disadvantage of Existing System

Existing methods of content protection, such as Digital Rights Management (DRM) and watermarking, may not provide adequate security against sophisticated piracy techniques, necessitating innovative solutions.

The unauthorized sharing and distribution of streaming content undermine the revenue streams of content creators and platform providers, leading to substantial financial losses.

As subscribers access content on OTT platforms, their personal information and viewing habits are at risk of exposure, raising concerns over privacy and data security.

C. Proposed System

The proposed system aims to enhance content protection on OTT platforms by leveraging the Least Significant Bit (LSB) method to embed data within video content streamed to users. This approach ensures robust data hiding while preserving the visual quality of the video. Our system operates by modifying the pixel data of the video frames, allowing for seamless encoding and decoding of data.

Least Significant Bit (LSB) Method: The LSB method involves replacing the least significant bit of each pixel's colour channel (RGB) with a bit of the data to be embedded. Since the LSB has the least impact on the pixel's colour intensity, this modification is imperceptible to the human eye. By sequentially replacing the LSBs of consecutive pixels in the video frames, we can embed data efficiently without significantly altering the video's visual appearance.

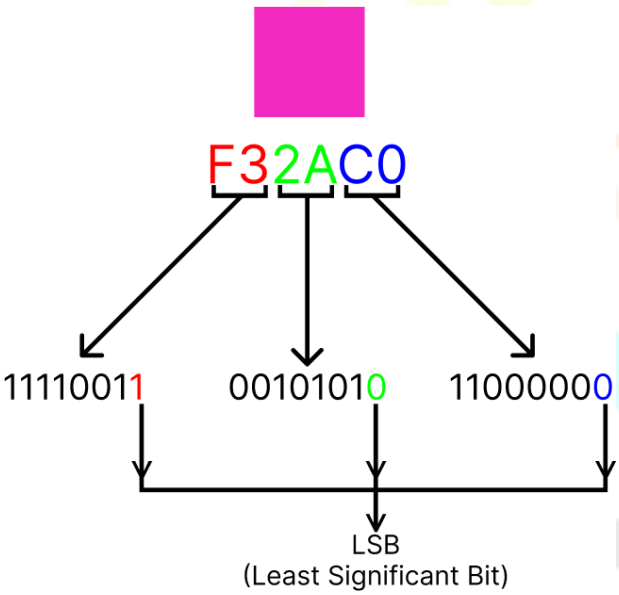

Fig 3.1 Least Significant Bit

Encoding Data in Video: To encode data into a video using the LSB method, we first extract the individual frames of the video. For each frame, we iterate through the pixel values and replace the LSBs with the corresponding bits of the data to be embedded. This process continues until all the data has been embedded into the video frames. The modified frames are then reassembled to reconstruct the video with the embedded data.

Decoding Data from Video: To decode data from a video encoded using the LSB method, we extract the first frame of the video, which contains the embedded data. We iterate through the pixel values of the frame and extract the LSBs to reconstruct the original data. This process allows us to recover the embedded data without requiring access to the entire video stream.

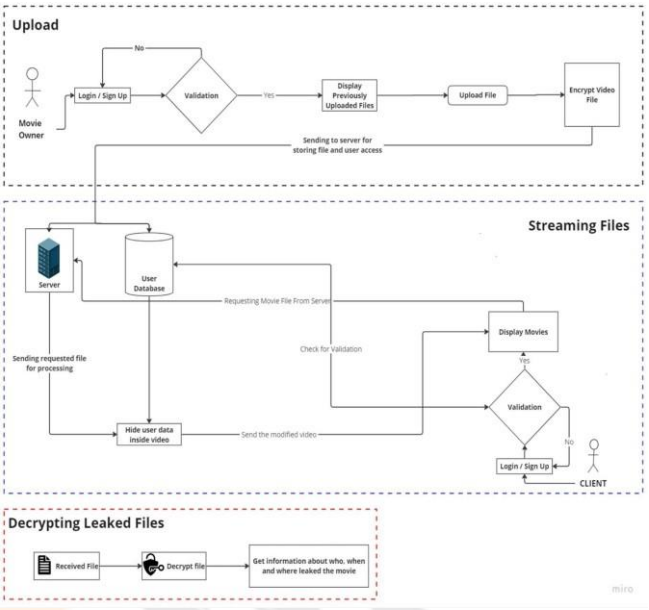

Fig 3.2 Architecture Diagram of the overall process

Detailed Steps:

- Frame Extraction: Extract the first frame of the video to access the embedded data.
- LSB Modification: Iterate through the pixel values of the frame and replace the LSBs with the corresponding bits of the embedded data.
- Data Extraction: Extract the modified LSBs from the pixel values to reconstruct the embedded data.

Advantages of the Proposed System**:**

- Robust Content Protection: The LSB method provides robust data hiding capabilities, ensuring that embedded data remains hidden from unauthorized users.
- Minimal Impact on Visual Quality: By modifying only the LSBs of pixel values, the proposed system minimizes the visual impact on the video content, preserving its quality.
- Efficient Encoding and Decoding: The encoding and decoding processes are efficient and can be performed in real-time, allowing for seamless integration into OTT platforms.

- Enhanced Security: By embedding data directly into the video frames, our system enhances content security and deters piracy attempts.

The proposed system offers a practical and effective solution for enhancing content protection on OTT platforms using the LSB method. By leveraging this method to embed data within video content, we ensure robust data hiding capabilities while preserving the visual quality of the video. Our system presents a versatile and adaptable approach to content protection, addressing the evolving challenges of piracy and unauthorized access in the digital entertainment landscape.

IV. System Requirements

A. Software Requirements

- *Python (Programming Language):*

  Python is the primary programming language used for developing the backend logic and data processing algorithms of the project. Used for implementing core functionalities, data manipulation, and integration with external libraries.

- *Django (Web Framework):*

  Django is a high-level Python web framework that facilitates rapid development and clean, pragmatic design of web applications. Utilized for building the web application's backend, handling HTTP requests, managing databases, and implementing server-side logic.

- *OpenCV (Computer Vision Library):*

  OpenCV is a popular open-source computer vision library that provides tools and algorithms for image and video processing. Employed for video manipulation tasks such as frame extraction, image processing, and computer vision-based operations.

- *MoviePy (Video Editing Library):*

  MoviePy is a Python library for video editing and manipulation, allowing for the creation, editing, and processing of video files. Used for video editing tasks such as video concatenation, trimming, watermarking, and embedding data into video streams.

- *HTML (Hypertext Markup Language):*

  HTML is the standard markup language used for creating the structure and content of web pages. Employed for designing the user interface (UI) and defining the structure of web pages, including layout, text, images, and forms.

- *CSS (Cascading Style Sheets):*

  CSS is a stylesheet language used for defining the presentation and visual appearance of HTML elements on web pages. Utilized for styling and formatting the UI elements created with HTML, including colors, fonts, margins, and layout.

- *JavaScript (Programming Language):*

  JavaScript is a scripting language used for adding interactivity and dynamic behavior to web pages. Implemented for client-side scripting, event handling, DOM manipulation, and asynchronous communication with the server.

- *jQuery (JavaScript Library):*

  jQuery is a fast, lightweight JavaScript library that simplifies HTML document traversing, event handling, and animation. Used for simplifying complex JavaScript tasks, including DOM manipulation, AJAX requests, and event handling, to enhance the responsiveness and interactivity of the web application.

- *Bootstrap (Front-end Framework):*

  Bootstrap is a popular front-end framework for building responsive and mobile-first web applications. Employed for designing responsive and visually appealing UI components, including grids, navigation bars, buttons, and forms, to ensure consistent and user-friendly layout across different devices and screen sizes.

- *Other Technologies:*

  Additional technologies are used depending on specific project requirements, such as database management systems, version control systems, and deployment platforms.

V. Module Description

A. Command Line Module

The Command Line Module serves as the entry point for interacting with the project's functionalities via the command line interface (CLI). It provides users with a convenient way to execute various commands and operations related to video manipulation, content protection, and account management.

Users can execute commands to manipulate videos, such as extracting frames, embedding data, concatenating videos, and applying filters or effects. User data is used to encrypt or decrypt videos, embed watermarks, and perform other content protection operations to safeguard intellectual property. This module includes robust error handling mechanisms to provide informative error messages and handle unexpected inputs or exceptions gracefully. The module also provides feedback to the user through console output, informing them of the status of their commands and

any errors encountered during execution. It also logs relevant information for debugging and monitoring purposes.

B. *Client Module*

The Client Module facilitates interaction between users and the project's functionalities through a graphical user interface (GUI). It provides an intuitive and user-friendly interface for accessing video manipulation, content protection, and account management features.

The module consists of graphical elements such as buttons, input fields, dropdown menus, and dialogs for user interaction. Users can view and playback videos within the GUI, with options for controlling playback speed, volume, and seeking. The module enables users to watch contents from the platform, perform account management actions directly from the GUI. The module provides feedback to users through status messages, progress bars, and notifications, ensuring a seamless and responsive user experience.

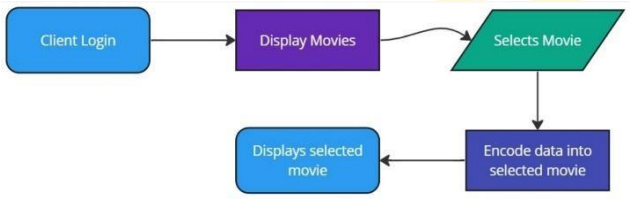

Fig 5.1 Flow chart of Client Module

It handles user interactions and events, such as button clicks, menu selections, and text input, to trigger the corresponding actions or operations. The module communicates with backend functionalities and modules to execute requested operations and retrieve data for display in the GUI. The Client Module includes error handling mechanisms to provide informative feedback and to handle unexpected situations.

C. *Movie Uploader Module*

The Movie Uploader Module is responsible for facilitating the uploading and processing of video files by movie creators. It provides a mechanism for movie creators to upload their videos to the platform, perform pre-processing tasks, and initiate content protection operations if required.

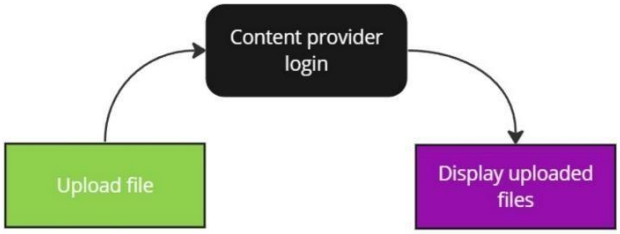

Fig 5.2 Flow chart of the Movie Uploader Module

The module may perform pre-processing tasks such as format conversion, resolution adjustment, or compression to optimize the videos for further processing. The module provides feedback on the upload progress and processing status, informing users of any errors or issues encountered during the process. The module handles file upload and management, including file selection, validation, and transfer to the server.

It also contains logic for performing pre-processing tasks on uploaded videos, such as invoking video manipulation functions or calling external libraries like OpenCV or MoviePy. To ensure security and access control, the module may incorporate movie uploader authentication mechanisms to verify the identity of users before allowing video uploads. For large video files or computationally intensive tasks, the module may employ asynchronous processing techniques to optimize performance and responsiveness.

D. *Account Management Module*

The Account Management Module facilitates user authentication, profile management, and access control functionalities within the project. It allows users to create and manage their accounts, update profile information, and control access to specific features or content.

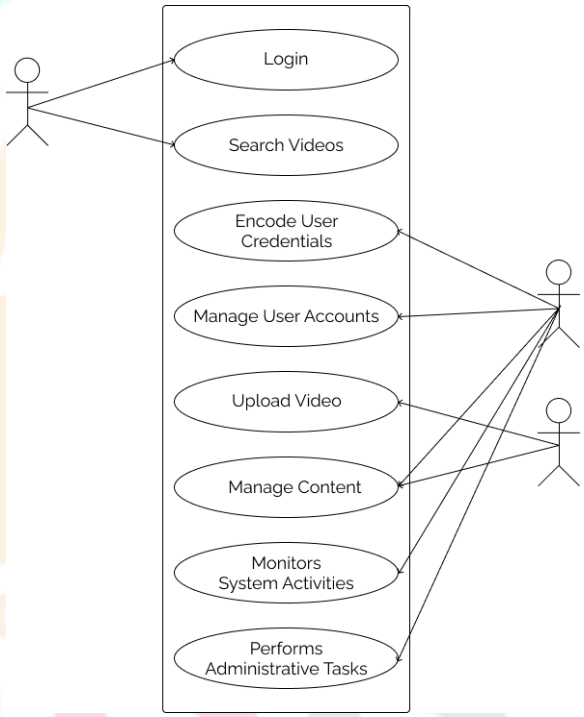

Fig 5.3 Use Case Diagram of the web application

VI. Result and Conclusion

The project successfully addresses the pervasive issue of piracy on OTT platforms through the implementation of a robust content protection solution. By leveraging advanced technologies and innovative approaches, we have developed a system that enhances content security, preserves user privacy, and mitigates the risks associated with unauthorized access and distribution of streaming content.

Our project represents a significant step forward in combating piracy and enhancing content protection on OTT platforms. Through the integration of cutting-edge technologies and a collaborative team effort, we have devised a comprehensive solution that addresses the challenges outlined in the problem definition. Our team's diverse skillsets and expertise in Python, Django, OpenCV,

MoviePy, HTML, CSS, JavaScript, jQuery, and Bootstrap have enabled us to efficiently utilize technology to solve real-world problems. Moving forward, we envision our solution making a tangible impact in the fight against piracy, safeguarding the interests of content creators, platform providers, and subscribers alike.

Our team comprises dedicated individuals with a passion for innovation and a commitment to excellence. Each member brings unique skillsets and expertise to the project, ranging from software development and web programming to data analysis and project management. With a collaborative mindset and a shared vision, we have worked seamlessly together to conceptualize, design, and implement the project, overcoming challenges and achieving milestones along the way.

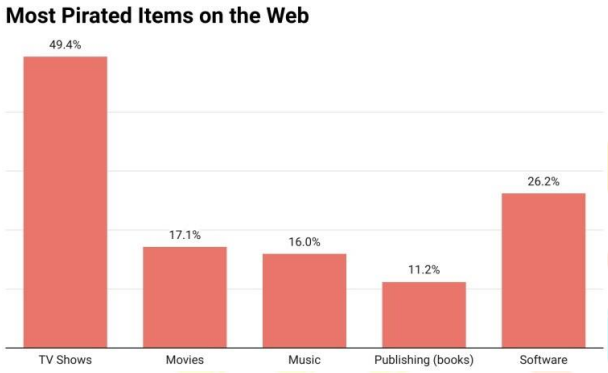

Fig 6.1 Statistics about most pirated content on Internet

According to industry reports, global revenue losses due to digital piracy are estimated to be in the billions of dollars annually. In light of these statistics, our project aims to address the significant impact of piracy on the digital entertainment industry by providing an effective and scalable solution for content protection on OTT platforms. Through ongoing collaboration and continuous improvement, we remain committed to advancing the fight against piracy and ensuring a secure and sustainable future for digital content distribution.

REFERENCES

[1] Niels Provos and Peter Honeyman, “Hide and Seek: An Introduction to Steganography”, University of Michigan, IEEE 2003

[2] Daniela Stanescu, Mircea Stratulat, Voicu Groza, Joana Ghergulescu and Daniel Borca, "Steganography in YUVcolor space", IEEE International Workshop on Robotic and Sensors Environments (ROSE 2007), Ottawa-Canada, pp.1-4, October 2007

[3] P.Ramesh Babu, Digital Image Processing. Scitech Publications., 2003.

[4] Johnson, N. F. and Jajodia, S. (1998). Exploring steganography: Seeing the unseen. Computer, 31(2): 26–34.

[5] Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett, “Steganography and digital watermarking” School of Computer Science, The University of Birmingham. 2003. www.cs.unibo.it/people/phdstudents/scacciag/home_files/teach/datahide.pdf.

[6] Debnath Bhattacharyya, P. Das, S. Mukherjee, D. Ganguly, S.K. Bandyopadhyay, Tai-hoon Kim, “A Secured Technique for Image Data Hiding”, Communications in Computer and Information Science, Springer, June, 2009, Vol. 29, pp. 151-159.

[7] Wang H., Wang S.,“Cyber warfare: Steganography vs. Steganalysis”, Communications of the ACM-Voting Systems,Vol. 47, No. 10, pp. 76-82, 2004.

[8] Chincholkar A.A. and Urkude D.A., "Design and Implementation of Image Steganography", Journal of Signal and Image Processing, ISSN: 0976-8882 & EISSN: 0976-8890, Volume 3, Issue 3, pp. 111-113, 2012

[9] Nagham Hamid, Abid Yahya, R. Badlishah Ahmad and Osamah M. Al - Qershi, "Image Steganography Techniques: an Overview", International Journal of Computer Science and Security (IJCSS), Volume (6) : Issue(3) : 2012

[10] Mamta Juneja, Parvinder S. Sandhu, and Ekta Walia,”Application of LSB Based Steganographic Technique for 8-bit Color Images”, WASET 2009

[11] Neeta Deshpande, Kamalapur Sneha, Daisy Jacobs,—Implementation of LSB Steganography and Its Evaluation for various Bits Digital Information Management, 2006 1st International

[12] Chia-Chi Wu, Chia-Chen Lin, Chin-Chen Chang, Digital rights management for multimedia content over 3G mobile networks, Expert Systems with Applications, Volume 37, Issue 10, 2010, ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2010.03.047.

[13] Arnab, Alapan & Hutchison, Andrew. (2004). Digital rights management-An overview of current challenges and solutions.

[14] Zhu, Bin. (2006). Multimedia Encryption. 10.1016/B978-012369476-8/50006-3.

[15] El-said, Shaimaa A., Khalid FA Hussein, and Mohamed M. Fouad. "Securing multimedia transmission over mobile communication channels." International Journal of Intelligent Engineering Informatics 1.3-4 (2011): 213-245.