A System to Preserve Metadata using Steganography

C.M. Kondasinghe

2017

A System to Preserve Metadata using Steganography

A dissertation submitted for the Degree of Master of Science in Information Security

C.M. Kondasinghe

University of Colombo School of Computing

2017

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: C.M. Kondasinghe

…….

Signature Date:

This is to certify that this thesis is based on the work of Ms. C.M. Kondasinghe under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by: Dr. Kasun de Zoysa

…….

Signature: Date:

Table of Content

List of Figures

List of Tables

Abstract

Metadata of a file is the data that provides information about other data. Metadata of a video describes the conditions under which the video was created such as date, time, camera model, resolution and geographical location. Since the metadata of a video is attached in separate files or in the header, current techniques make the metadata unsecure and easy to get lost, removed or modified. As a result, various researches have been done to address this issue in multiple mechanisms including watermarking, steganography and sensor patterns. Through this research author analyses existing mechanisms to secure metadata. As for the analysis of existing security mechanisms author justifies the need to implement a system using steganography and cryptography mechanisms on securing metadata. Steganography is used to hide information based on the fact that the existence of information is invisible to the user and cryptography is used with the purpose of enhancing security of hidden data. Then author discusses the design of the metadata preserving system and moves on to evaluating the system via comparison of parameters of original video and stego video. Through the evaluation author was able to demonstrate the performance of the system.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Kasun de Zoysa for the continuous support during my MSc study and related research, for his patience, motivation, and immense knowledge. His guidance helped me during the time of research and writing of this thesis.

Besides my advisor, I would like to thank all the lecturers who were presented for my proposal and interim defense for their insightful comments and encouragement, and also for hard questions which incented me to widen my research from various perspectives.

Last but not least, I would also like to thank the collogues of my MSc class, my friends and my family: my parents and to my husband for supporting me spiritually throughout writing this thesis and in my life in general.

Chapter 1: Introduction

Metadata is a data that gives information about other data. The metadata of a particular file can give information about its characteristics, quality, creator information, time/date of creation, purpose of creation, procedure of creation, geographical location and the characteristic of the hardware [3].

There are many incidents and clues that explain the importance of metadata. As the advent of the Unmanned Aerial Vehicles (UAVs) has been increased, the protection of the information within the transmitted or stored video has become a big challenge. Most known drone systems attach metadata of the recorded video in separate files or in the header of the video. Current techniques make the metadata unsecure and easy to get lost and removed as well as it occupies more storage and bandwidth [4]. Another incident shows about arresting of anonymous hacker. FBI has arrested an anonymous hacker by using a picture that has been posted by him. The metadata of that picture contain GPS (Global Positioning System) information of where the picture had been taken [3].

There are number of tools and utilities to find, modify or remove metadata of original source such as ExifTool, ExifTool Linux, Exif Viewer and FOCA Online [3]. Modification of original metadata of a file may lead to provide wrong information about the incident which is revealed via that file. Hence researches have been done to preserve metadata in different theoretical backgrounds such as using watermarking, steganography and sensor patterns depending on the application. However, most of the available works still lack providing high payload for protected data and mostly the data is hidden as a plain text using steganography.

Steganography is the art of covered or hidden writing. The purpose of steganography is covert communication to hide a message from a third party [1]. The medium which the message is hidden is called carrier or cover object in such a way that existence of the message is concealed. The cover object could be a digital still image, an audio file, or a video file. The hidden message called payload could be a plain text, an audio file, a video file, or an image [2]. The difference between cryptography and steganography is that steganography involves hiding information so it appears that no information is hidden at all. If a person or persons views the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information [1].

Steganographic methods can be classified into spatial domain embedding and frequency domain embedding. Least Significant Bit (LSB) replacing is the most widely used steganographic method in spatial domain, which replaces the cover image's LSBs with message bits directly. LSB steganography is a popular method because of its low computational complexity and high embedding capacity [2].

In this research, author introduces a mechanism to preserve metadata using steganography and further uses cryptography to encrypt the hidden data for the purpose of enhancing security.

1.1 Motivation

The information about a video is stored in it's metadata. Keeping the security of the metadata of a video has become a challenge, since metadata is easy to get modified or removed with current techniques. Different methodologies have been proposed to secure metadata such as watermarking, steganography and sensor patterns. Studying about hiding secret messages, steganography is used together with cryptography in order to enhance security, but lesser number of researches has been done for the purpose of securing metadata. Therefore the researcher decided to introduce a system to preserve metadata using steganography, and to use cryptography as a further security enhancement.

1.2 Research Question

Since metadata of a file includes the descriptive, structural and administrative information, metadata is beneficial in discovery and identification of the particular file. But with upcoming techniques, metadata is subjected to tampering and can be removed. Different technologies have been proposed for the purpose of preservation of metadata. Via this research the researcher addresses the question that, how steganography and cryptography can be used to preserve metadata of a video.

1.3 Objective

The objective of this research is to design and implement a system to preserve metadata using steganography and cryptography.

1.4 Scope

1. Identify issues and problems associated with security of image/video metadata.
2. Analyze existing security mechanisms designed for preserving metadata and identify the issues associated with existing mechanisms.
3. Analyze existing algorithms of steganography and other data hiding mechanisms.
4. Design a suitable system for metadata preservation of a video using data hiding mechanisms.
5. Implementation of system.
6. Performance analysis of metadata preserving system.

1.5 Structure of Thesis

The rest of the thesis is organized as follows.

Chapter 2 – Literature Review

This chapter identifies similar solutions and research work done in the area of study and helps to identify the requirements and the limitations in detail.

Chapter 3 – Design

This chapter describes the design technique background and design steps of the system.

Chapter 4 – Implementation

This chapter discusses about the implementation of the method. All components including functions used for the implementation and functionalities of the built system are identified and described.

Chapter 5 – Evaluation

This chapter describes the methodologies used to verify and validate the approach and explains how the aspects of the objective were tested.

Chapter 6 – Conclusion and Future Work

This chapter concludes the dissertation by discussing the final results and the future works.

Chapter 2: Literature Review

2.1 Overview

With the evolution of Information technology and communication, information security has become an important fact on Internet domain [5]. This research proposes a solution for the problem of modifying metadata of videos using steganography and cryptography in order to enhance security of hidden message.

Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography [5].

Following Sections study these approaches in depth.

2.2 Importance of Metadata

Metadata is the data that provides information about other data. Three distinct types of metadata exist, namely descriptive metadata, structural metadata and administrative metadata. Descriptive metadata describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords. Structural metadata is metadata about containers of data and indicates how compound objects are put together, for example, how pages are ordered to form chapters. It describes the types, versions, relationships and other characteristics of digital materials. Administrative metadata provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it [18].

Metadata has become a powerful tool to organize and search through the growing libraries of image, audio and video content that users are producing and consuming. This is especially important in the area of digital photography where, despite the increased quality and quantity of sensor elements, it is not currently practical to organize and query images based only on the millions of image pixels. Instead, it is best to use metadata properties that describe what the photo represents and where, when and how the image was taken. Metadata is now critical in workflows ranging from consumer sharing experiences to professional-level asset management [19].

Forensic image analysis has been the main driver of the field but also video files have recently been brought to the forefront. Similar to differences in the JPEG file structure, manufacturer- and model-specific video file format characteristics and point to traces left by processing software [20].

Now a days as most of the videos are transmitted through the wireless media which can sometimes be used as a proof of evidence but due to the advanced technologies which are used for manipulating the videos and images it has become very essential to authenticate the videos or the images. Authenticity is provided so that if any of these videos or images is considered as the proof of evidence it must be considered as valid proof [38]. In [38] the

author has used the IP address, system information, video information, MAC address to identify the source of the video.

It is increasingly common for file metadata to include the Global Positioning System (GPS) coordinates denoting the actual location where the recording was created. Depending on the GPS reception and capabilities of the recording unit, the GPS coordinates can pinpoint a specific location where the video was created, for example a room of a high-rise building. The file's metadata may include the date-time of the file's creation and who was the last person to access it. Metadata may also include details about the software or equipment used to capture the recording, including the user settings in effect at the time of file saving. The metadata can be subjected to tampering [30]. In [31] Marcinak and Mobasseri explain that UAVs collect voluminous amount of metadata along with surveillance footage. Metadata stream is transmitted on a separate channel or stored in the header section of video, which leaves metadata in the open and easily removable [31].

2.3 Cryptography, Steganography and Watermarking

In order to build the approach to preserve metadata, a review was done on different hiding mechanisms.

Steganography, Cryptography and Watermarking are well known and widely used to hide the original message. Steganography is used to embed message within another object known as a cover work, by tweaking its properties. By using Cryptography sender convert plaintext to cipher text by using Encryption key and other side receiver decrypt cipher text to plain text. Digital watermarking is a technique for inserting information (the watermark) into an image (visible or invisible). [6]

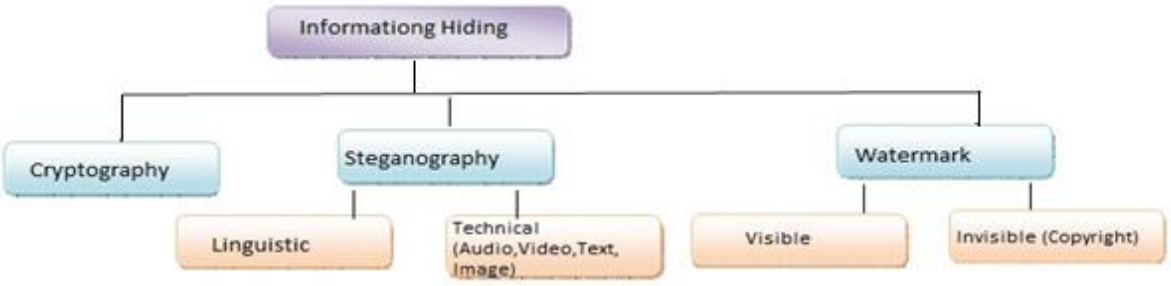

Figure 2.1: Information Hiding Mechanisms [6]

2.3.1 Cryptography

Cryptography means sender convert plain text to cipher text by using Encryption key and other side receiver decrypt cipher text to plain text by Decryption Key. The idea is to change the text in to format which is not easy to decrypt without decryption key. [6]

The Sender encrypts the data with a key and converts the text in to cipher text which is a scrambled text. This cipher text is transmitted at the receiver end. The receiver decrypts the

data with the key and gets the original text. Cryptography schemes include private key cryptography, public key cryptography and hash functions. [7]

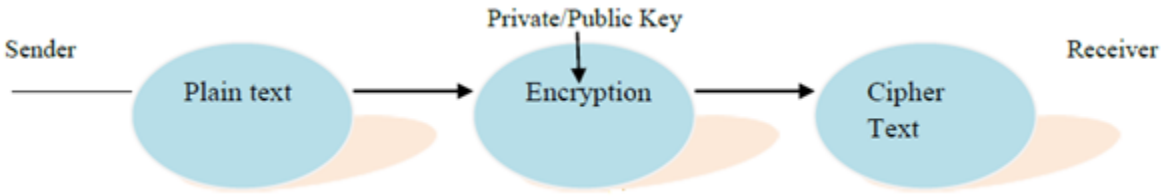

Figure 2.2: Process of Encryption [7]

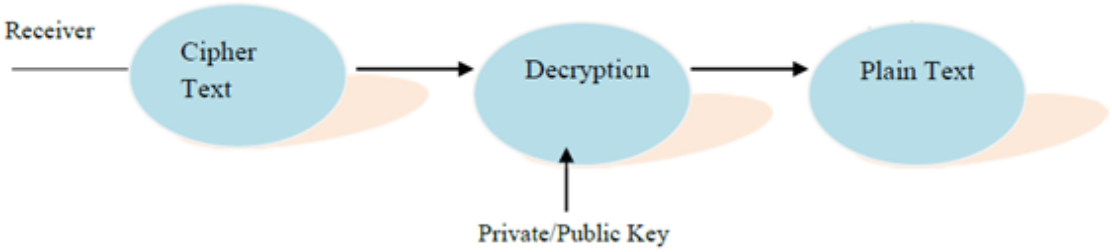

Figure 2.3: Process of Decryption [7]

2.3.2 Steganography

Steganography is the art or practice of concealing a file, message, image, or video within another file, message, Image or video. The steganography technique takes a cover image secret data and a key, embeds the secret data into the cover image and produce a stego image. This stego image is transferred to the receiver end and the secret message is extracted by the recipient if he knows the key [7].

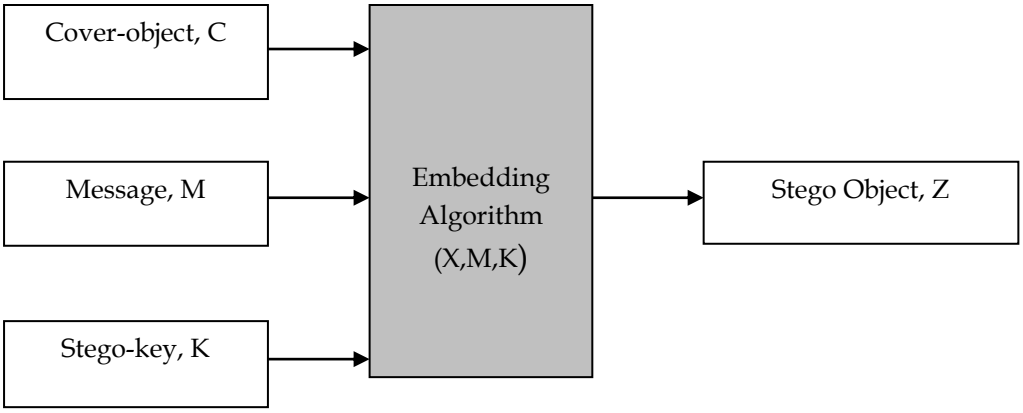

Figure 2.4: Process of Steganography [7]

2.3.3 Watermarking

Digital watermarking is the act of hiding a message related to a digital signal (i.e. an image, song, video) within the signal itself [32]. Explaining further, the process of inserting information (the watermark) in the carrier file (either visible or invisible form) is termed as digital watermarking. [7]

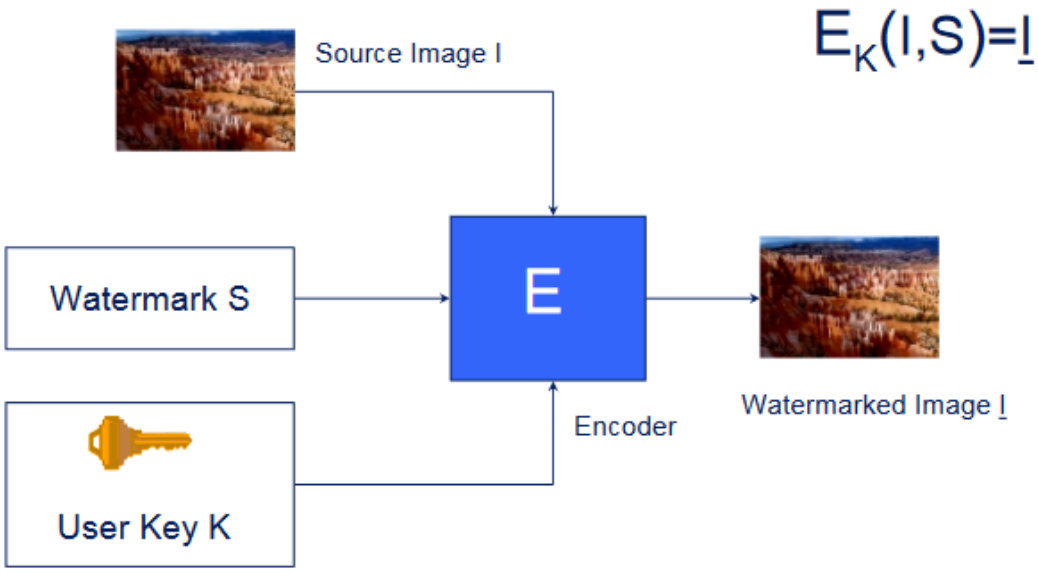

Figure 2.5: Process of Watermark Encoding [8]

No	Attributes	Steganography	Cryptography	Watermarking
1	Definition	Steganography is called as cover writing	Cryptography is termed as hidden secret	A kind of marker covertly embedded in image or audio
2	Techniques	LSB , Spatial, Block complexity, Transform Domain	Transposition, substitution, Stream ciphers, Block ciphers	Spatial domain, Fragile watermarking
3	Naked eye identification	No, as message is Hide within other carrier (cover image)	Yes, as message is convert in other way, which sough something is hidden	Yes, as actual message is hiding by some watermark
4	Applicable	Cosmically	Cosmically	Cosmically
5	Type of attack	Steganalysis i.e. if the intruder detects that steganography is performed then the security breaks	Cryptanalysis i.e. if the intruder cracks the cipher text then the security is broken as the original message is revealed	Watermark Drowning, synchronization attacks, stochastic attacks
6	Secret Key	May be used	Necessary cannot work without key	May be used
7	Motive	Conceal the existence of message	Conceal the contents of the message not its presence	Copyright protection
8	Outcome	Stego image	Cipher text	Watermarked image
9	Durability	Steganography basically hides the data under a cover i.e. it does not make any changes to the data	Cryptography , using an encryption algorithm converts the plain text in to cipher text i.e. it makes changes to the original data	Watermarking embed the data covertly in to the noise signals

Table 2.1: Comparison between Steganography, Cryptography and Watermarking [7, 8]

Studying the different data hiding mechanisms, for the purpose of preserving metadata steganography was identified as the most suitable method based on the facts that, cryptography conceals the message, converted in other way and the observers can identify that something is hidden. In watermarking the watermark is seen by observers, where it is exactly suitable for copyright purposes. For the purpose of preserving metadata by hiding it inside the carrier itself and also to fulfill the requirement of invisibility, steganography was chose as the hiding mechanism.

2.4 Steganography

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out to the usual [41].
In general, the information hiding process extracts redundant bits from cover-object. The process consists of two steps [41].

- Identification of redundant bits in a cover-object. Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the cover-object.

- Embedding process then selects the subset of the redundant bits to be replaced with data from a secret message. The stego-object is created by replacing the selected redundant bits with message bits.

2.4.1 Steganography in an image

To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in "noisy" areas that draw less attention, those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. A number of ways exist to hide information in digital media. Common approaches include below methods[1].

• Least significant bit insertion
• Masking and filtering
• Redundant Pattern Encoding
• Encrypt and Scatter
• Algorithms and transformations

Each of these techniques can be applied, with varying degrees of success [1].

2.4.2 Steganography in an audio file

In a computer-based audio steganography system, secret messages are embedded in digital sound. The secret message is embedded by slightly altering the binary sequence of a sound file. Existing audio steganography software can embed messages in WAV, AU, and even MP3 sound files [9].
Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media, such as digital images. In order to conceal secret messages successfully, a variety of methods for embedding information in digital audio have been introduced. These methods range from rather simple algorithms that insert information in the form of signal noise to more powerful methods that exploit sophisticated signal processing techniques to hide information. The list of methods that are commonly used for audio steganography are listed and discussed below. [1]

• LSB coding
• Parity coding
• Phase coding
• Spread spectrum
• Echo hiding

2.4.3 Steganography in a video file

Video files are generally a collection of images and sounds, so most of the presented techniques on images and audio can be applied to video files too [10]. The great advantages of video are the large amount of data that can be hidden inside and the fact that it is a moving stream of images and sounds. Therefore, any small but otherwise noticeable distortions might go by unobserved by humans because of the continuous flow of information [11, 12].

2.4.4 Embedding a text file into a video file

Ramalingam [14] had developed a stereo machined to develop a steganography application to hide data containing text in a computer video file and retrieve the hidden information. This was done by embedding a text file into a video file in such a way that the video does not lose its functionality using Least Significant Bit (LSB).

2.4.5 Steganography mechanisms

Reviewing on different steganography mechanisms, Least significant bit (LSB) insertion is a common and simple approach to embed information in an image file. Masking and filtering techniques are mostly used on 24 bit and grey scale images. They hide info in a way similar to watermarks on actual paper and are sometimes used as digital watermarks. Masking images entails changing the luminance of the masked area. The smaller the luminance change, the less of a chance that it can be detected [1]. Redundant pattern encoding is used to scatter hidden information throughout the cover images ("patchwork" is a method that marks image areas, or patches) [15]. The DCT (Discrete Cosine Transform) is a mathematical transform that takes a signal and transforms it from spatial domain into frequency domain. DCT coefficients are used for JPEG compression. It separates the image into parts of differing importance. It transforms a signal or image from the spatial domain to frequency domain. It can separate the image into high, middle and low frequency components [16].

Analyzing the attributes of different steganography mechanisms, LSB method can afford high payload capacity of secret message [41]. Also LSB results with comparatively higher PSNR (Peak Signal to Noise Ratio) and lower MSE (Mean Squared Error) values. Comparatively DCT is higher in invisibility. Anyhow in DCT and DWT (Discrete Wavelet Transform) the payload capacity is lower compared to LSB. Also the PSNR in DCT and DWT is comparatively lower than in LSB [17]. Analyzing the features of different steganography mechanisms, LSB was identified as the most suitable method for the particular research question on metadata preserving, based on the facts of high payload and PSNR. Metadata of a video includes set of details (secret message in the research) and after applying cryptography to the secret message there is a considerable change in size. Therefore high payload capacity of LSB mechanism is considered as a major advantage.

2.5 Video file formats

Beginning from the home video era up to the most cutting-edge standards of today, video file formats have undergone some major changes. Different file formats do different things, and the right video format for a specific file isn't necessarily the right one for the others. Each file format has its own set of specifics. A normal video file in a digital format is made up of two parts, a "codec" and a "container". A "codec" is used to compress and decompress a video file, as there are times where video files are too large and may cause difficulty when trying to download or play the file. Some examples of "codecs" are FFMpeg, DivX, XviD, and

x264. A “container” is a collection of files that stores information about the digital file. It simply means there is a combination of both audio and video data in a single file to allow for simultaneous audio-with-video playback. Some popular types of “containers” are AVI, FLV, WMV, MP4, and MOV [21]. In this research, the researcher tends to select one type of video format (.avi) for the system implementation considering to the time constrains.

2.5.1 AVI format (.avi)

Microsoft introduced AVI (Audio Video Interleave) in 1992 as a multimedia container format, which can contain both video and audio streams. AVI files start with a mandatory AVI RIFF (Resource Interchange File Format) header. All following data is organized and stored in so-called lists and chunks. A four character code (FourCC) is used to identify these data segments. The FourCC for a list is LIST. Different FourCC’s exist for chunks, e.g., JUNK or idx1. There is no strict specification that defines the sequence and occurrence of lists and chunks [20]. Figure 2.6 elaborates the file structure of AVI type.

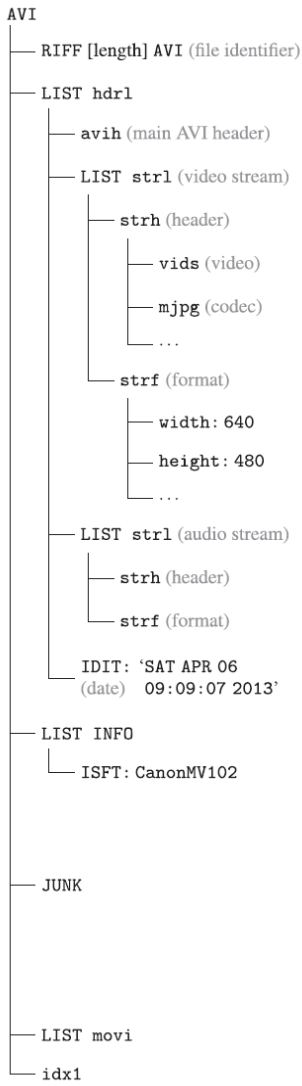

Figure 2.6: AVI file structure [20]

2.6 Encryption mechanisms

Encryption can be additionally used to increase security of the hidden message in the stego image. Below table shows a comparison of Symmetric Key Algorithms. [27], [28]

Attribute	**DES**	**3DES**	**AES**	**BLOWFISH**
Designers [27]	IBM	IBM	Joan Daemen& Vincent Rijmen	Bruce Schneier
First Published [27]	1977	1998	1998	1993
Key size [27]	56 bits	112/168 bits	128/ 192/ 256 bits	32-448 bit in steps of 8 bits. 128 bits by default
Security [27]	In secure	Secure than DES	Secure	Secure
Speed [27]	Fast	Slow than DES	Fast	Fast
Encryption [27]	High	Moderate	High	High
Features [28]	Not structure, Enough	Adequate security	Replacement for DES, Excellent Security	Excellent security

Table 2.2: Comparison of Symmetric Key Algorithms

It was decided to use Rijendael algorithm as the cryptographic method to encrypt the secret message considering high security and speed.

2.6.1 Rijndael encryption algorithm

The Rijndael algorithm is a block cipher which means that for a given encryption key, the algorithm performs a series of transformations on a block of data. The same encryption key can be used to reverse these transformations in the process of decryption. Rijndael was selected as the Advanced Encryption Standard (AES) in 2001 by the National Institute of Standards and Technology (NIST). [26]

The Rijndael algorithm is an iterative block cipher. It has a variable key and block length, which gives it a degree of flexibility when choosing an appropriate implementation. The block and key sizes can be any multiple of 32 bits between 128 bits and 256 bits. This is the key difference between the Rijndael algorithm and the version of it specified as the AES, as the AES fixes the block length at 128 bits and key sizes of 128, 192or 256 bits. [26]

2.7 Related research on preserving metadata using hiding mechanisms

As digital media is easily transported, copied or exchanged, a marked file provides innumerable advantages for tracking, ownership verification and covert communication. UAVs collect voluminous amount of metadata along with surveillance footage. Presently, metadata stream is transmitted on a separate channel or stored in the header section of video, where metadata is in the open and easily removable [31]. Author of [31] present an application of video digital watermarking to metadata embedding in UAV video footage. In their research, they proposed sending the metadata information within the video by embedding the metadata using digital watermarking. Their method used VLC (Variable Length Codes) representation to build the pair trees where the watermark bits are embedded by either changing the specific bits in some blocks to indicate embedding '1' or leaving the block unchanged to indicate embedding '0'. Anyhow VLC approach in watermarking is now considered obsolete and researchers in this field have stopped using this method due to the absence of contributions offered to improve this mechanism [4].

In images, digital watermarking has been used by [33] to embed the metadata of taken photos by the digital camera. The associated data such as resolution, date, time, and more were encoded in their algorithm and embedded into the image based on DCT domain. Another algorithm was proposed by [34]. In this algorithm, the DWT with three levels of decomposition was used. In this result, the Low (PSNR) was achieved about 35. Another method for video watermarking that used DCT was proposed by [35]. In their method, DCT was obtained from each frame for the luminance part only after converting the video into the YUV color space. The embedding of the watermark bits was executed diagonally in the coefficients in each block obtained from 8 × 8 blocks of the DCT transformation. This algorithm showed low PSNR value less than 35.

Another Algorithm was proposed in [36] to for data hiding in video files which was intended as steganography method. In this algorithm, Kanade-Lucas-Tomasi (KLT) tracking was executed using Hamming codes to track the facial regions and encode the secret message. LSB were used as the main method of embedding the bits on the spatial domain for the chosen regions pixels. Authors of this algorithm tried to increase the visual quality parameter. High PSNR values were attained. But their results showed very low detection accuracy reached 0.760. Author mentions the requirement to improve payload in future work.

A real-time video watermarking algorithm for surveillance networks was proposed by [37]. In this algorithm, the video frames were segmented into candidate-scenes-based after transforming the frames into DCT domain. These selected scenes are obtained by finding the relationships between the frames. The VLC run level pairs are changed to embed the bits in the macro blocks. The experiments showed an acceptable PSNR value of 42 dB. However, the bit-rate has a slight decrease and there is no evaluation on checking maximum payload.

Digital forensics deals with the process of uncovering and interpreting electronic data. The goal of the process is to preserve any evidence in its most original form while performing a structured investigation by collecting, identifying and validating the digital information for the purpose of reconstructing past events [38]. The advance digital technologies have brought us numerous cheap yet good-quality imaging devices to capture into the digital videos, so it became a big challenge to identify the source of video which are used as a proof of evidence in forensic field. In [38] the author has presented a mechanism to input the IP address, system information, video information and MAC address into the video using LSB in order to identify video source.

In [39] the author had used the wavelet based technique to identify the source. Camera source details are a part of metadata. Peak-to-Correlation Energy (PCE) criteria is used to check the correlation performance. The advantage of the method described here is that it works for almost all type of digital camera. In [40] the author had focused on building a classifier to effectively distinguish between digital images taken from digital single lens reflex (DSLR) and compact cameras. Source camera class identification scheme for DSLR and compact Cameras is proposed based on machine learning classifiers utilizing statistical features of wavelet sub-bands and noise residues.

According to the literature review it was identified that there exist a high risk in metadata modification and therefore a solution is proposed via this research using steganography. Further symmetric cryptography is used to encrypt the metadata for enhance the security.

Chapter 3: Design

3.1 Design Background

This section discuss about the background necessary to understand the design concept of the system. The section contains background information on video metadata structure and extraction, steganography mechanism, mechanism on encrypting the secret message.

3.1.1 AVI file structure

There are 3 types of AVI files:

- AVI 1.0: The original, old AVI file.
- Open-DML:An extension to the AVI file format. Overhead is reduced.
  Hybride-Files: Open-DML les that contain an additional Legacy Index for compatibility reasons. [25]

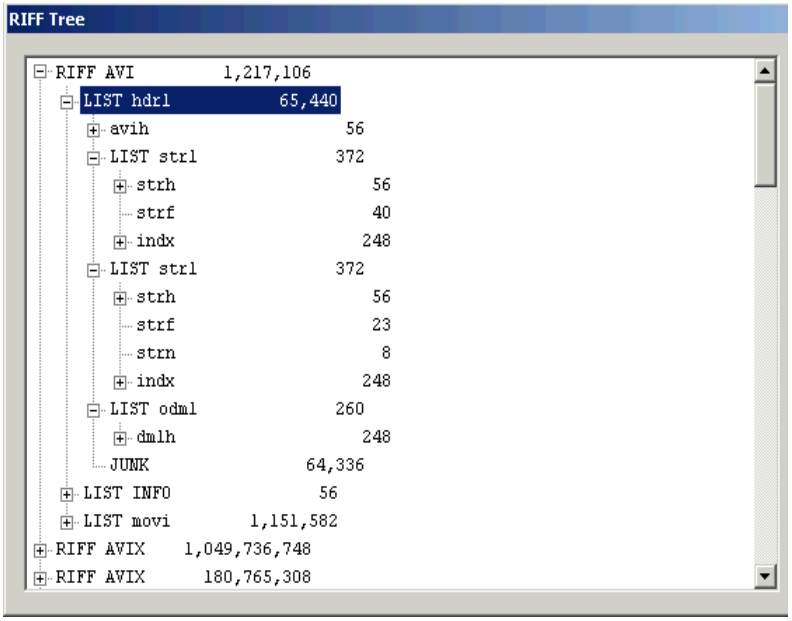

Figure 3.1: AVI header

Structure of avih (main header) is as follows. [25]

- dwMicroSecPerFrame: Contains the duration of one video frame in microseconds.
- dwMaxBytesPerSec: Highest occurring data rate within the file.
- dwTotalFrames: Contains the number of video frames in the RIFF-AVI list
- dwStreams: Number of streams in the file
- dwSuggestedBufferSize: Size of buffer required to hold chunks of the file.
- dwWidth: Width of video stream
- dwHeight: Height of video stream

3.1.2 Least Significant Bit (LSB) steganography

As per the literature review, the researcher selects Least Significant Bit (LSB) steganography as the mechanism for data hiding, mainly based on high payload capacity. Least Significant Bit (LSB) insertion is an approach for embedding information in a carrier file. The simplest steganography techniques embed the bits of the message directly into least significant bit plane of the cover-image in a deterministic sequence [41]. Modulating the least significant bit does not result in human perceptible difference because the amplitude of the change is small. In this technique, the embedding capacity can be increased by using two or more least significant bits [41]. This technique works good for image, audio and video steganography and to the human eye, the resulting image will look identical to the cover object. [1]

Explaining the theory behind LSB methodology, if we consider image steganography then the letter A can be hidden in three pixels (assuming no compression). The original raster data for 3 pixels (9 bytes) may be,
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

The binary value for A is 10000001. Inserting the binary value for A in the three pixels would result in,

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)
The underlined bits are the only three actually changed in the 8 bytes used. On average, LSB requires that only half the bits in an image be changed. You can hide data in the least and second least significant bits and still the human eye would not be able to discern it. [1] Figure 3.2 elaborates the LSB mechanism.

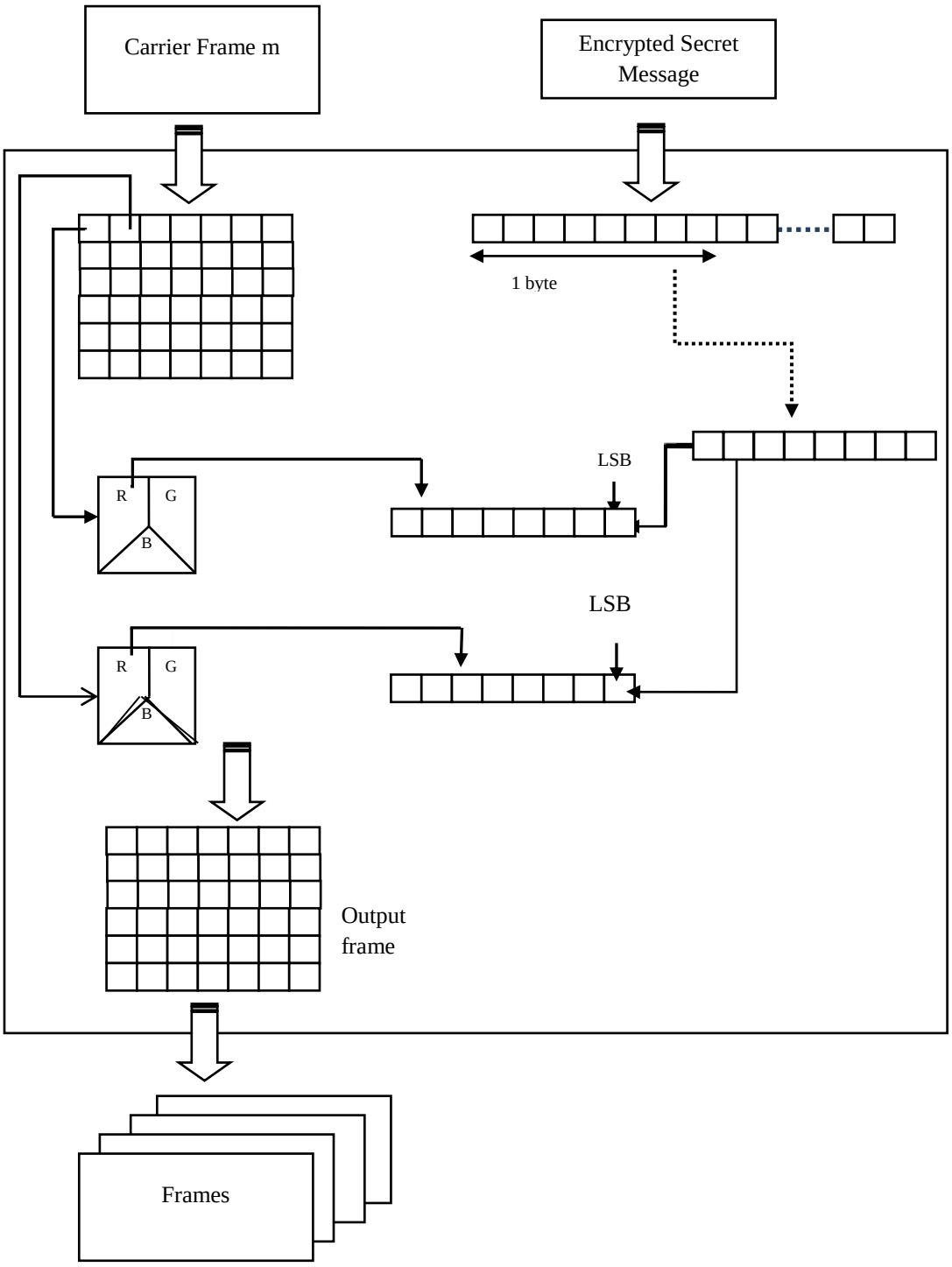

Figure 3.2: Embedding a Message stream in Carrier using LSB

3.2 Design

The proposed design of the system shall be described using following steps.

1. Extract metadata from the selected video.
2. Split frames of the video.
3. Secure metadata using encryption.
4. Embed secured metadata into selected frames using LSB.
5. Combine frames into output Video.

As depicted in figure 3.3, the metadata of the input video is extracted as the first step. The frames are split from the input video. The extracted metadata is encrypted via Rijndael algorithm. Encrypted metadata is hidden the split frames using LSB method. To find the relevant pixels, an algorithm is used which will be based on variable ASCII values of a key defined by the user. The frames which are embedded with secured hidden message (metadata of the carrier file itself) are combined and the output video is created.

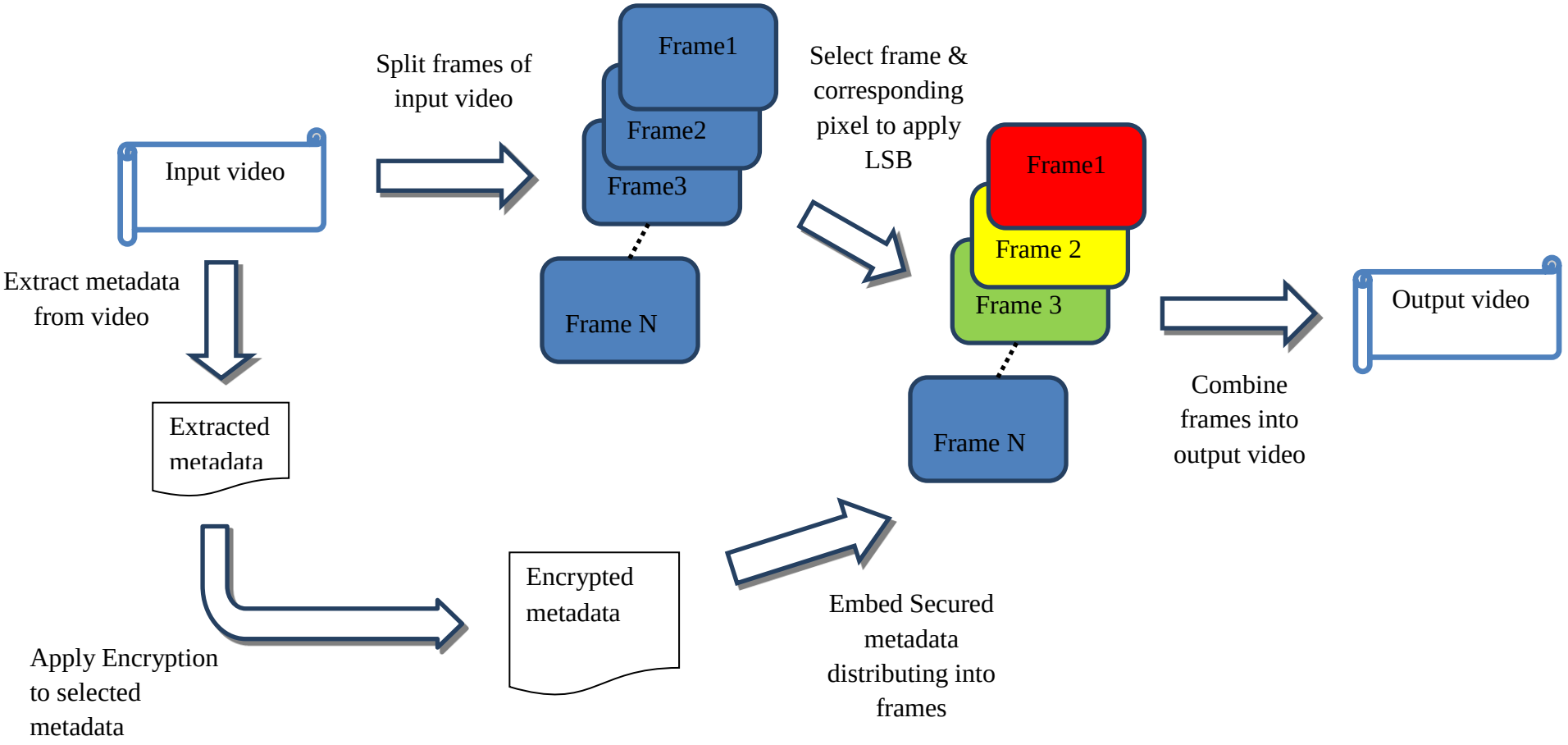

Figure 3.3: Architecture of Proposed System

3.2.1 Extract metadata

AVI contains 3 parts file header, blocks (chunks), and index as depicted in figure 3.4 [22].

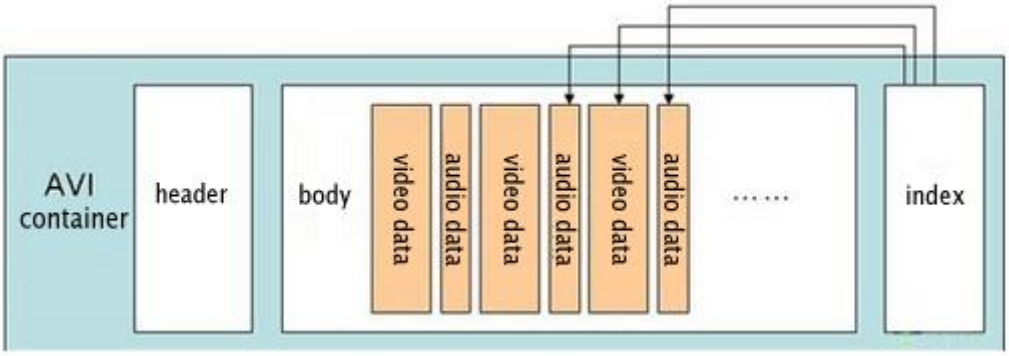

Figure 3.4: AVI file structure

Header contains metadata about the video [22] [42].

1. Date created
2. Date last modified
3. File Length
4. Frame rate : Number of Frames per second
5. Frame size : Frame width into Frame size
6. File format
7. File size
8. Bit rate mode
9. Bit rate
10. Pixel width
11. Pixel Height
12. GPS location
13. Display aspect ratio
14. Frame rate mode
15. Resolution
16. Bits/(pixel x Frame)
17. Stream size

According to the user requirement, any of the metadata or all metadata can be secured by using the system. Out of the whole details found in metadata of the video, the top five attributes in the above would be considered for the purpose of hiding in the implementation. This can be customized as per user requirement.

3.2.2 Splitting frames of the video

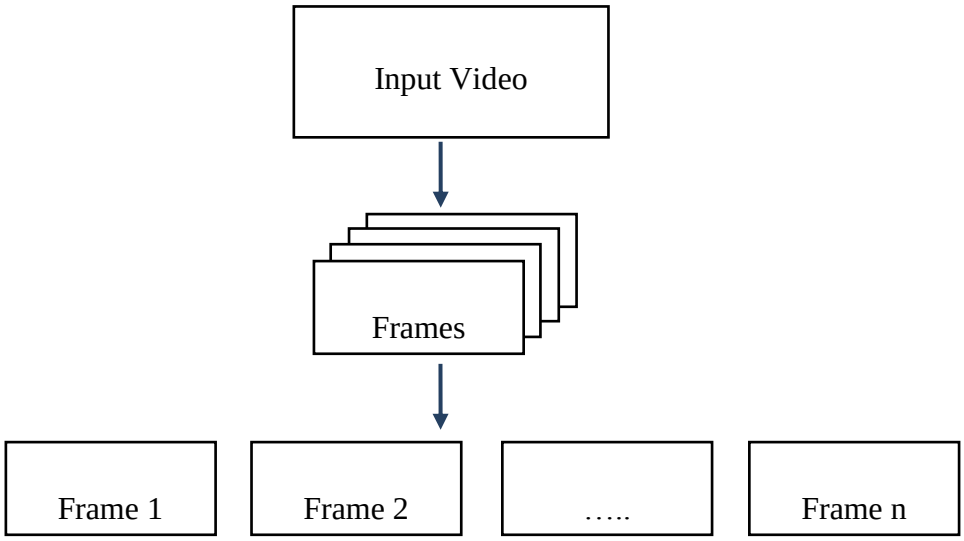

Figure 3.5: Frame splitting

In the proposed approach the frames would be extracted from the video in order to proceed with next steps (figure 3.5).

Frame rate (expressed in frames per second or fps) is the frequency (rate) at which consecutive images called frames are displayed in an animated display. The term applies equally to video. [23]

3.2.3 Secure metadata using encryption

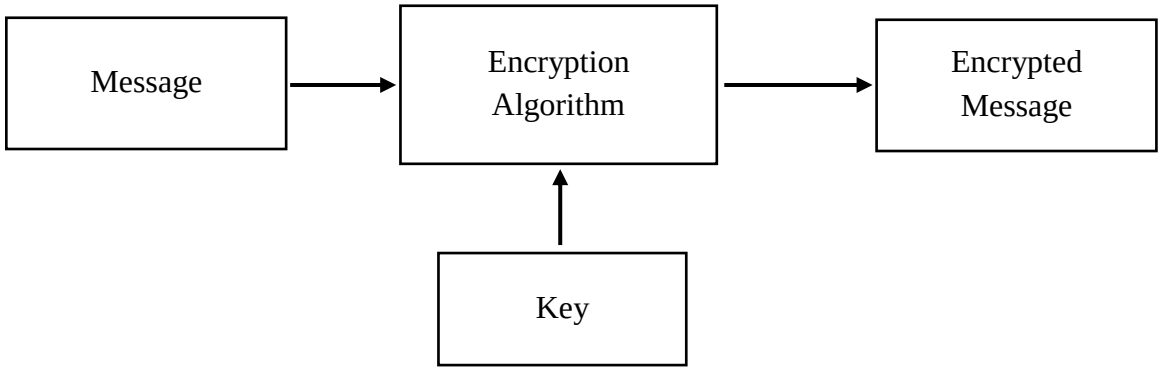

Figure 3.6: Encrypting the secret message

The extracted metadata (in 3.1.1) should be secured before hiding in carrier video. Encryption is used for this purpose. Rijndael encryption algorithm is used for this. Protected key is given by the user.

3.2.4 Embed metadata into frames using steganography

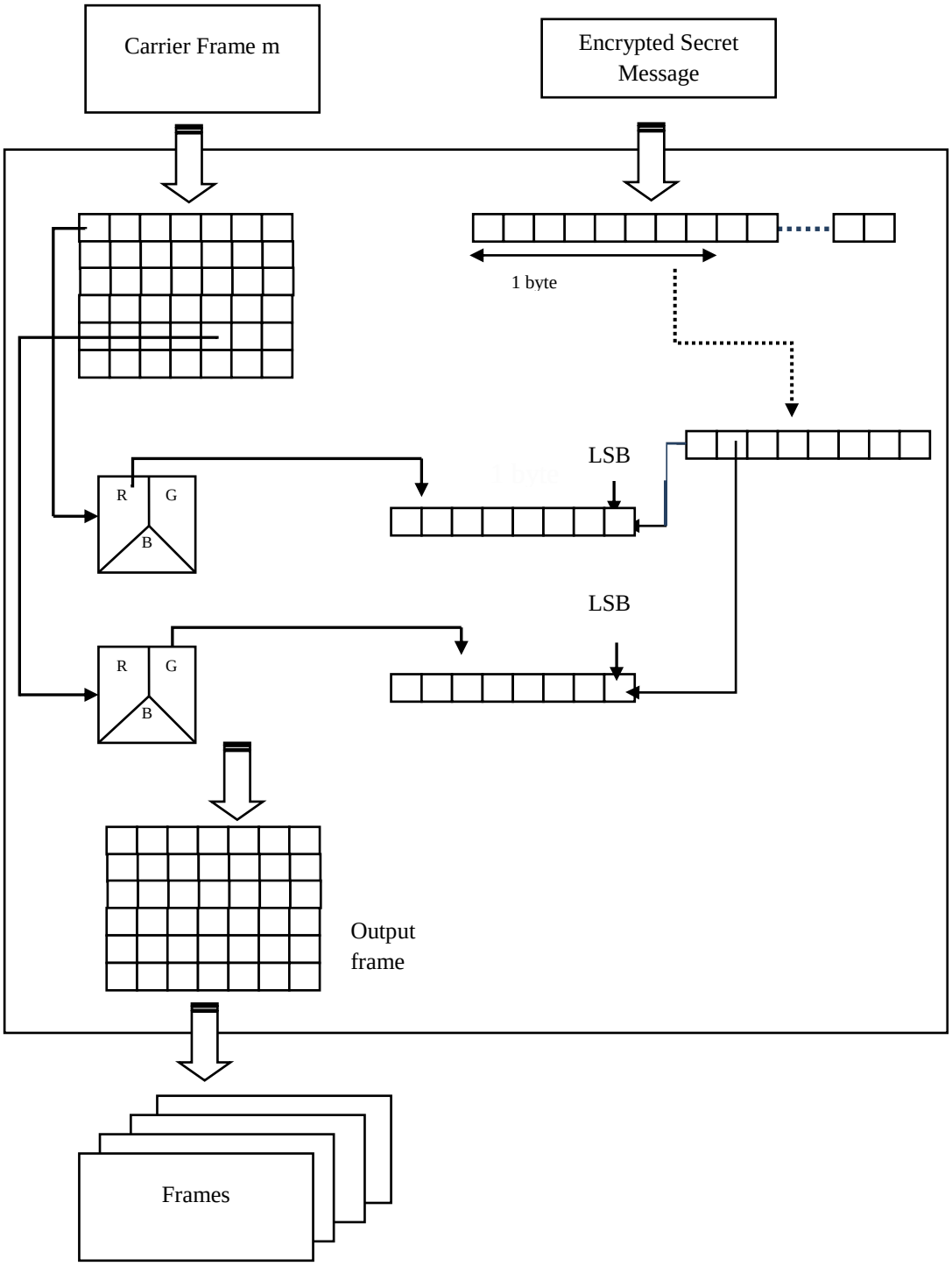

Figure 3.7: Hiding the encrypted message using LSB method

Steganography mechanism used for embedding secured metadata to video would be LSB (Least Significant Bit). The message would be not hidden in one frame, but be hidden in several frames in a distributed way in order to enhance security.
Below information are used to calculate the relevant parameters in hiding mechanism.

Frame Width: Width of one frame extracted from video file (in pixels)
Frame Height: Height of one frame extracted from video file
Frame Rate: Number of Frames per time unit
Length: Duration of video file

Two parameters, namely 'Carrier Units' and 'Usable Carrier Units' are calculated in order to find pixels to be used for hiding.
'Carrier Units' gives the total number of available pixels in all the frames in the video.

Carrier Units = Frame Width * Frame Height * Frame Rate * Length

In order to select the pixels which the secret message to be hidden, a key is defined which will be defined by the user. Count of 'Usable Carrier Units' depends on a key given by user. Initially (before uploading Carrier file) a key is given in order to choose the pixel which is used to apply LSB.

Usable Carrier Units = Carrier Units/ Average of ASCII values of key

The advantage having an additional key is, the positions that the secret message will be hidden in Carrier will vary according the ASCII of given key. As shown in figure 3.7, the least significant bit of the pixel (which the message bits would be hidden) is selected based on the user given key, and at one instance, the least bit of one color component in selected pixel (R,G,B) would be replaced by a message bit (bit of metadata stream in this scenario).

3.2.5 Generate stego video

After frames are embedded with parts of secret message the final step is to combine all frames & generate the output stego video, which would be also in AVI format.

Chapter 4: Implementation

4.1 Overview

The system was needed to implement using suitable methods. The performance of the system depends on the system design as well as the language and the libraries which are used for the system. This chapter describes the implementation and use of the system.

4.2 Embedding of secret message

The system is implemented with two key elements, hiding the secret and extracting. Interface of hiding the message is shown in figure 4.1.

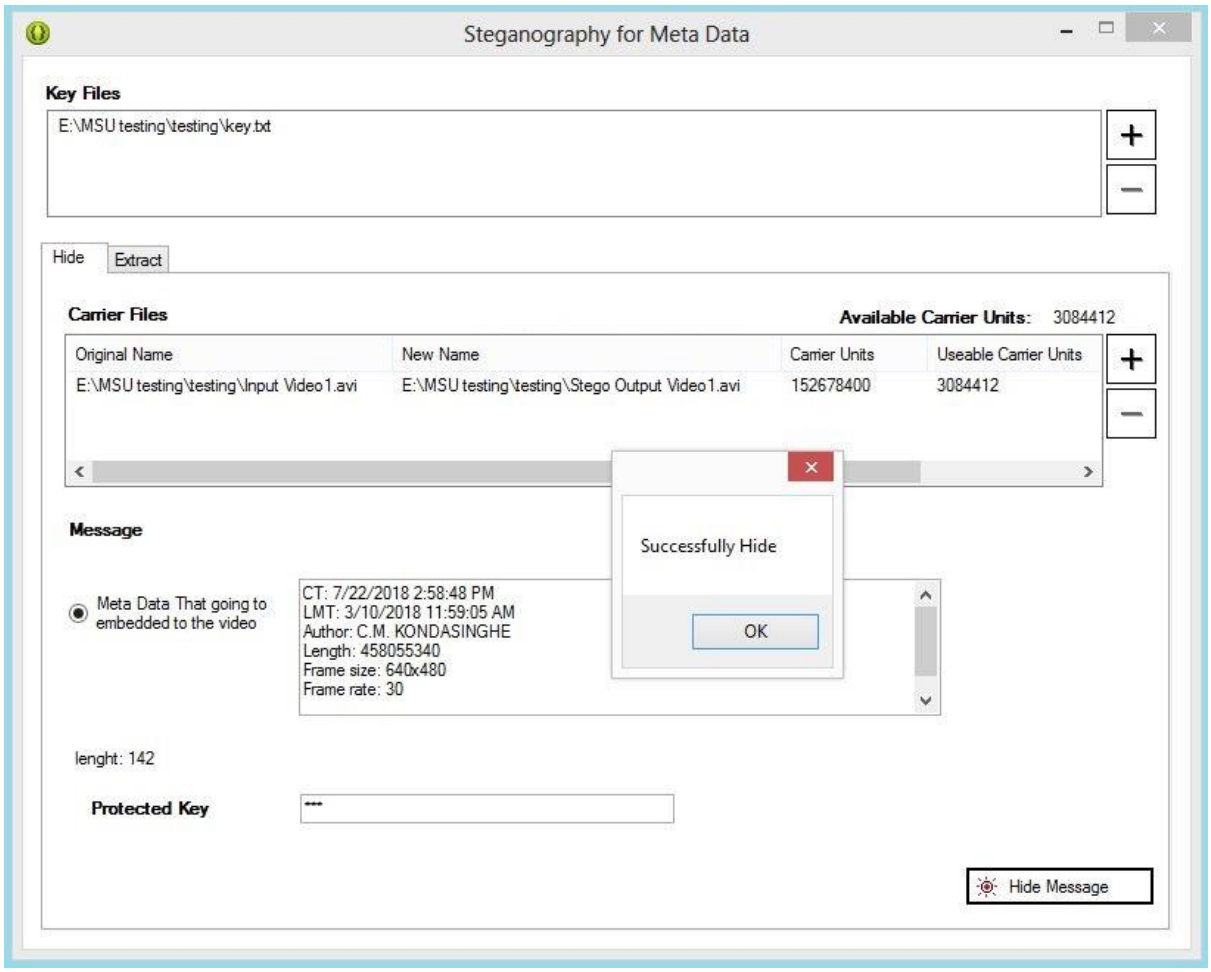

Figure 4.1: Interface for hiding

As inputs to the system, the key file with ASCII values is uploaded to the tab 'Key files' as shown in figure 4.1. ASCII values in this file are used to select the pixels which the message is hidden. The original video is uploaded to the tab 'Carrier Files' (Input Video 1.avi in figure 4.1). The location of stego file to be created is given as 'Stego Output Video1.avi'. Below functions were used in the process of hiding secret.

GetFileInfo()

This function is used to get file information from the source video.
Additionally to the extracted details, information of the owner is embedded as the information inside the secret message. This can be customized as required by owner of the video.

EncryptText()

Secret message (extracted video metadata and added author details) is encrypted via the function, using Rijndael algorithm. Encryption key is a variable which is given by the user. Key size is defined as 256 bit key.

CountUnits()

Total number of available pixels in all the frames in the original video is calculated using the function and displayed as Carrier Units in the interface.

CountUseableUnits()

The parameter 'Usable Carrier units' is calculated which is required to define the pixels to be used for secret message hiding. This is based on ASCII values of given key by the user.

GetColorComponent()

At each instance one color component out of (R,G,B) is used to replace with LSB. To get relevant color component of (R,G,B) a rotation method was used.

SetBit()

Least significant bit of the selected pixel is replaced with the message bits.

4.2.1 Video frame splitting and generation

The avifil32.dll with C # language is used in this approach. Accessed functions of avifil32.dll for the purpose of frame splitting and generation of carrier are as follows [24].

AVIFileOpen

The AVIFileOpen function opens an AVI file and returns the address of a file interface used to access it. The AVIFile library maintains a count of the number of times a file is opened, but not the number of times it was released.

AVIFileGetStream

The AVIFileGetStream function returns the address of a stream interface that is associated with a specified AVI file.

AVIStreamStart

The AVIStreamStart function returns the starting sample number for the stream.

AVIStreamLength

The AVIStreamLength function returns the length of the stream.

AVIStreamInfo

The AVIStreamInfo function obtains stream header information.

AVIStreamGetFrameOpen

The AVIStreamGetFrameOpen function prepares to decompress video frames from the specified video stream.

AVIStreamGetFrame

The AVIStreamGetFrame function returns the address of a decompressed video frame.

AVIFileCreateStream

The AVIFileCreateStream function creates a new stream in an existing file and creates an interface to the new stream.

AVIStreamSetFormat

The AVIStreamSetFormat function sets the format of a stream at the specified position.

AVIStreamWrite

The AVIStreamWrite function writes data to a stream.

AVIStreamGetFrameClose

The AVIStreamGetFrameClose function releases resources used to decompress video frames.

AVIStreamRelease

The AVIStreamRelease function decrements the reference count of an AVI stream interface handle, and closes the stream if the count reaches zero.

AVIFileRelease

The AVIFileRelease function decrements the reference count of an AVI file interface handle and closes the file if the count reaches zero.

AVIFileExit

The AVIFileExit function exits the AVIFile library and decrements the reference count for the library.

4.3 Extraction of secret message

Properties of the stego video contains it's original information as metadata as well as the hidden secret. As specified in the literature review the metadata can be modified using various tools. The advantage of this system is even if the metadata is removed or modified, the original information could be revealed from the stego video by extracting hidden message.

Figure 4.2 shows the interface of extracting the hidden message. The stego video (Stego Output Video1.avi referring to figure 4.1), same key containing ASCII values and the same encryption key should be input to the system for the message extraction. For the demonstration, the metadata of Stego Output Video1.avi is modified using the tool 'Attribute changer'. Even the metadata file is modified, the original information is extracted through the system as shown in figure 4.2. In figure 4.2 the real information can be viewed under 'Extracted Meta Data Text' while modified wrong information is viewed under 'Current Video Meta Data'.

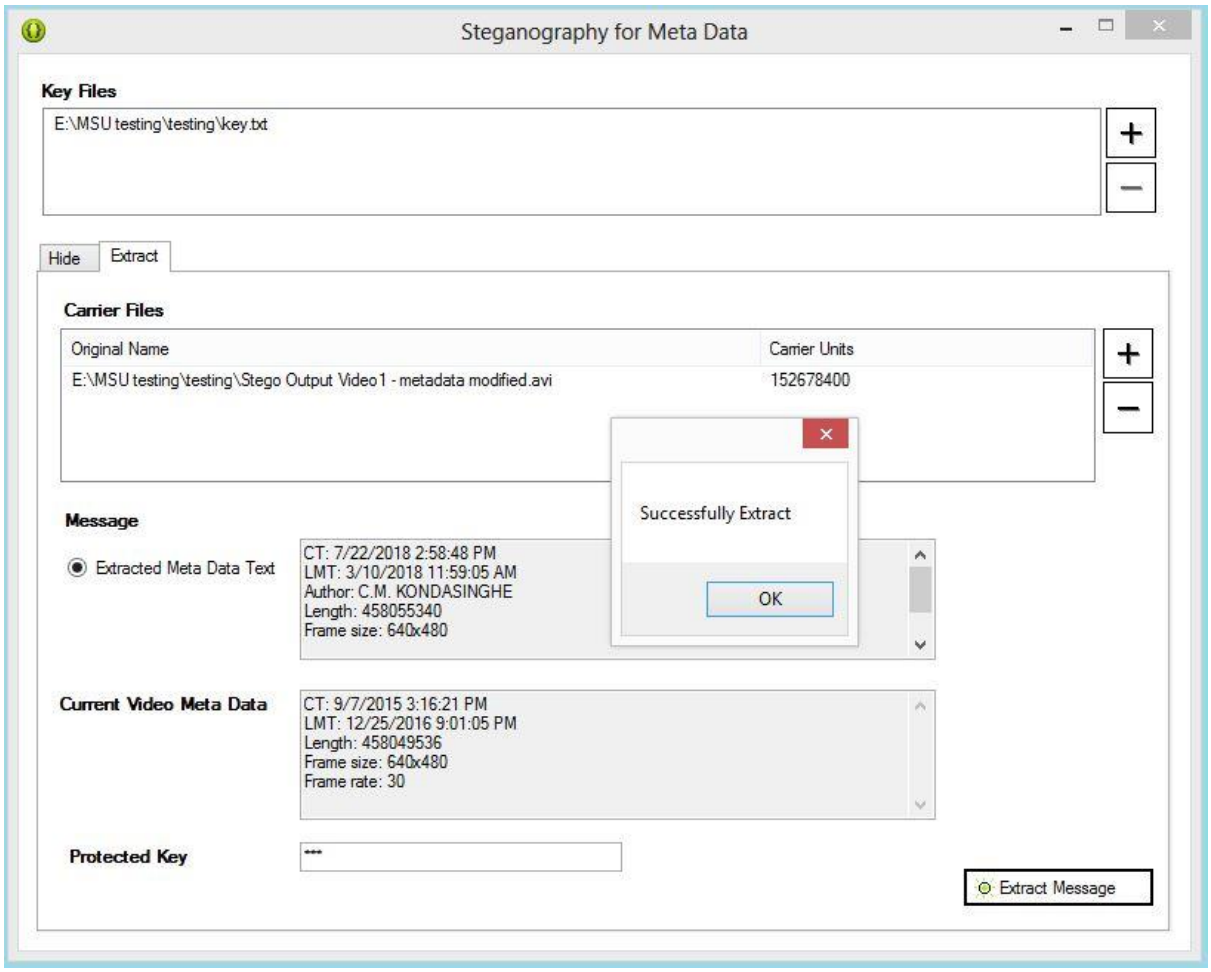

Figure 4.2: Interface for extracting hidden message

Chapter 5: Evaluation

5.1 Availability of hidden data after modifying metadata

The capability to extract the hidden data after the actual metadata is modified is checked in this evaluation method. Possibility to alter or remove original metadata is the major motivation to design a system to preserve metadata. Therefore in this method, alteration of true metadata is checked for a sample video taken from the designed software. The secured video which metadata is securely embedded is referred in the common name template 'Stego Output Video x'. 'Attribute Changer 9' was used as the metadata modifying tool, which is a powerful freeware used to change attributes of original files. After the alteration, the metadata is observed in general procedure and it is observed that metadata on general properties are changed. Then the altered video is processed though the designed system and it is observed that the true metadata exists within the video as a hidden secret.

Step 1:

Observe properties of stego video (which metadata of original video is embedded into).
Sample video: Stego Output Video1.avi

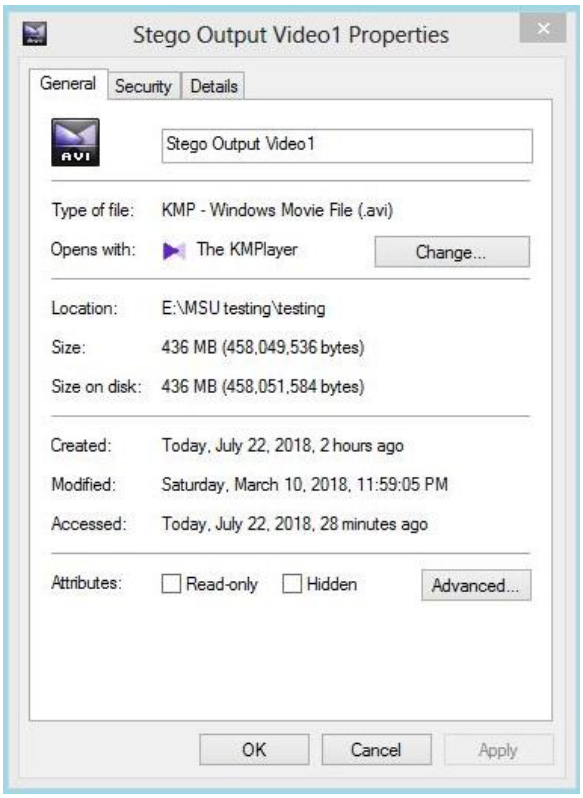

Figure 5.1: Properties of Stego Output Video1

Step 2:

Change Metadata of embedded video using the metadata modifying tool ‘Attribute changer 9’ [43].

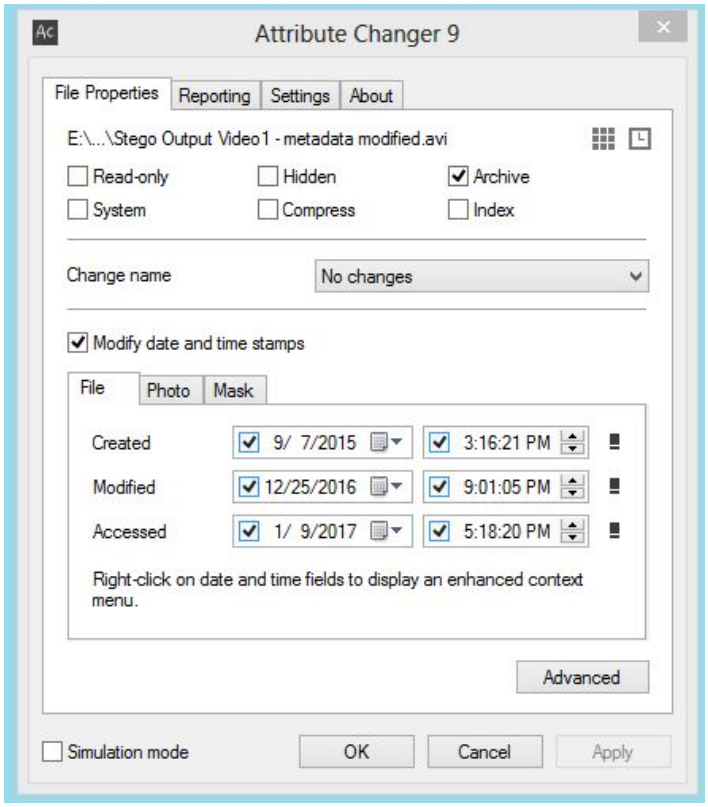

Figure 5.2: Change metadata using a software

Step 3:

Observe metadata of embedded video via properties.

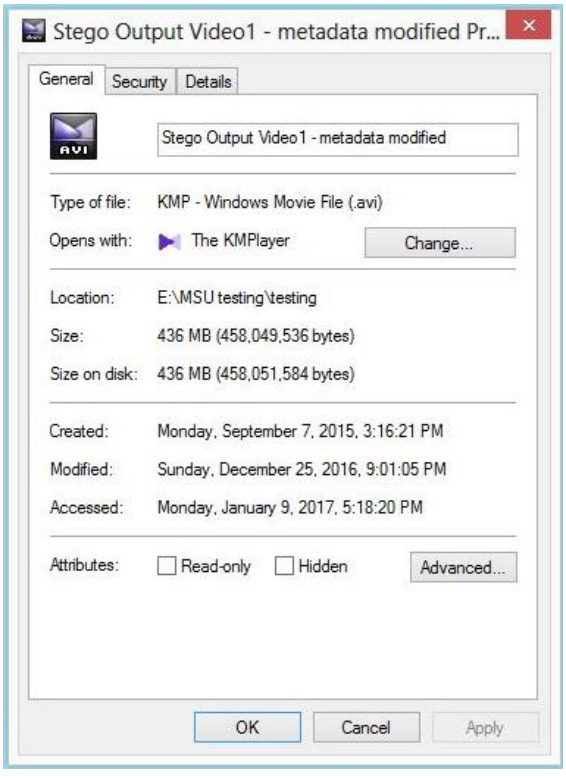

Figure 5.3: Properties of modified video

Step 4:

Insert this video into software and extract hidden data from embedded video. True metadata exists as the hidden message.

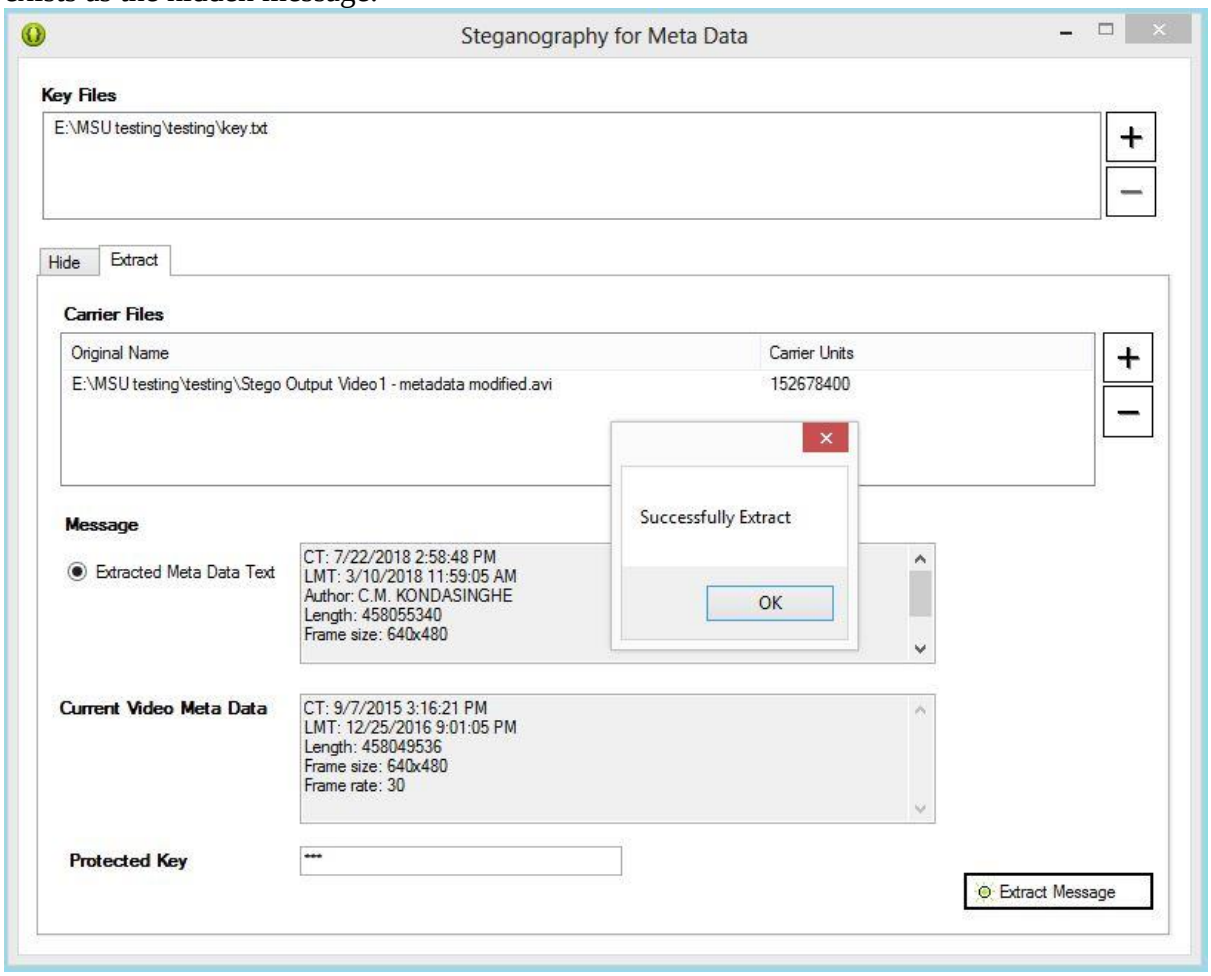

Figure 5.4: Extracting true metadata from designed system

5.2 Size of input video vs stego output video

Ratio between size of Stego output video and original input video for different sample encryption keys is checked in this evaluation criteria.

Sample Video	Size of input video (KB)	Size of stego output video (KB)	Ratio between output/input	Encryption key
Input Video1	447,320.00	447,314.00	99.9987%	1
Input Video2	482,421.00	482,415.00	99.9988%	1
Input Video3	488,721.00	488,715.00	99.9988%	1

Input Video4	440,120.00	440,114.00	99.9986%	1
Input Video5	498,622.00	498,615.00	99.9986%	1
Input Video6	428,420.00	428,414.00	99.9986%	1
Input Video7	423,920.00	423,914.00	99.9986%	1
Input Video8	473,421.00	473,415.00	99.9987%	1
Input Video9	348,318.00	348,312.00	99.9983%	1

Table 5.1: Ratio of size between input and output videos for Encryption key=1

Sample Video	**Size of input video (KB)**	**Size of stego output video (KB)**	**Ratio between output/input**	**Encryption key**
Input Video1	447,320.00	447,314.00	99.9987%	cha@12
Input Video2	482,421.00	482,415.00	99.9988%	cha@12
Input Video3	488,721.00	488,715.00	99.9988%	cha@12
Input Video4	440,120.00	440,114.00	99.9986%	cha@12
Input Video5	498,622.00	498,615.00	99.9986%	cha@12
Input Video6	428,420.00	428,414.00	99.9986%	cha@12
Input Video7	423,920.00	423,914.00	99.9986%	cha@12
Input Video8	473,421.00	473,415.00	99.9987%	cha@12
Input Video9	348,318.00	348,312.00	99.9983%	cha@12

Table 5.2: Ratio of size between input and output videos for Encryption key= cha@12

5.3 Variation of hiding pixel location according to key

The pixels used to hide data will vary for every video according to the given key (not a constant always).
Method: Get the position of pixels used to hide data keeping the video constant.

Key	**Hiding pixel point variation due to various keys**
123	(x=49 y=0), (x=99 y=0), (x=150 y=0), (x=199 y=0), (x=249 y=0), (x=300 y=0),..
abc	(x=97 y=0), (x=195 y=0), (x=294 y=0), (x=391 y=0), (x=489 y=0), (x=588 y=0),..

ucsc@10	(x=117 y=0), (x=216 y=0), (x=331 y=0), (x=430 y=0), (x=494 y=0), (x=543 y=0), (x=591 y=0),..
2013mis014	(x=50 y=0), (x=98 y=0), (x=147 y=0), (x=198 y=0), (x=307 y=0), (x=412 y=0), (x=527 y=0),..

Table 5.3: Variation of Hiding pixel point according to ASCII value of given Key

5.4 PSNR of stego videos

PSNR (Peak Signal to Noise Ratio) of a video is defined as follows.
PSNR is defined via the Mean Squared Error (*MSE*). Given a noise-free *m*×*n* monochrome image *I* and its noisy approximation *K*, *MSE* is defined as follows [29].

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

The PSNR (in dB) is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Here, MAXI is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255 [29]. The acceptable PSNR value is expected to be greater than 38 dB [45].
PSNR of sample videos were calculated using the tool MSU Video Quality measurement tool [44]. Figures in Appendix B depict the PSNR sample output videos where it was observed PSNR for all samples were over 90 dB and it is proved that PSNR is within acceptable range.

5.5 Performance of hiding the existence

In this method performance is evaluated by getting the probability of visibility of existence of the message to the human eye. To achieve this sample set of investigators were used to detect whether there is a difference between original videos and embedded videos. Investigator is given the opportunity to observe videos separately or both simultaneously running on same screen. The results are shown in below table.

	Video 1	**Video 2**	**Video 3**
Investigator 1	N	N	N
Investigator 2	N	N	Y
Investigator 3	N	N	N
Investigator 4	N	N	N
Investigator 5	N	N	N
Investigator 6	N	N	N
Investigator 7	N	N	N

Table 5.4: Response of investigators on visibility of difference for sample videos

Success ratio of clarity of Embedded video = (20/21)%

= 95.24%

Chapter 6: Conclusion and Future Work

6.1 Overview

This chapter contains the general discussion of the study. The discussion, findings and the conclusion are briefly outlined in this chapter. It includes the limitations and further suggestions in order to improve the research.

6.2 General Discussion

In the field on information security, people tend to interest about steganography in past years. The advantage of steganography is that only the message will be hidden in cryptography where as in steganography, the existence of the message can be hidden.
The current study was done in order to hide a secret message in a video (carrier). The difference with most other steganography projects is that researcher uses the metadata of the own video as the secret message. The reason to hide metadata is, metadata contains important details about the video and there are many online and free tools to modify/alter metadata. Therefore researcher tries to give an additional security for metadata by hiding it in video itself where the existence is not exposed. Additionally before applying steganography, the metadata stream is encrypted with Rijndael encryption algorithm.

6.3 Findings and capabilities of developed solution

The researcher firstly tried to get metadata & preserve frame-wise, but lately found metadata of each frame is not separately saved, but as a unique chunk. Therefore the metadata was extracted as a whole and then hidden in divided frames.
Developed solution is based on LSB mechanism. But every least significant bit of a frame is not replaced with message stream bits. To choose the pixel which is used to hide data, a key is used. Therefore the original video is not highly distorted due to process. Also there's no significant change in size of original and embedded video. By encrypting the secret message before hiding, the robustness is increased.

6.4 Limitations

For a video file, the audio and video files are needed to separate before applying steganography. Due to the lack of time, this separation part was neglected. Therefore, only visual files can be used as carrier. Next limitation of the study is, only uncompressed Microsoft avi visuals can be used as carrier.

6.5 Future Work

As future improvement the mechanism can be developed to be applied for other compressed video formats and also for videos containing audio part as well. Without restricting to a pre-defined encryption algorithm the user can be given flexibility to choose encryption mechanism and key size.

References

1. Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee and Poulami Das, "A Tutorial Review on Steganography", 2008.
2. Lifang Yu, Yao Zhao, Rongrong Ni, Ting Li, "Improved Adaptive LSB Steganography Based on Chaos and Genetic Algorithm", 2010.
3. Irfan Shakeel, 2013, "Metadata and Information Security", [Online] Available: http://resources.infosecinstitute.com/metadata-and-information-security/#gref
4. Nasr addin Ahmed Salem Al-Maweri, Aznul Qalid Md Sabri, Ali Mohammed Mansoor, Unaizah Hanum Obaidellah, Erma Rahayu Mohd Faizal, Joan Lai P., "Metadata hiding for UAV video based on digital watermarking in DWT transform", 2016.
5. T. Morkel 1, J.H.P. Eloff 2, M.S. Olivier 3, "An overview of image steganography", 2005.
6. Hardikkumar V. Desai, "Steganography, Cryptography, Watermarking: A Comparative Study", Journal of Global Research in Computer Science, Volume 3, No. 12, 2012.
7. Latika, "A Comparative Study of Cryptography, Steganography & Watermarking", Journal of Emerging Technologies and Innovative Research (JETIR), Volume 2, Issue 5, 2015.
8. Nasir Memon, "Information Hiding, Digital Watermarking and Steganography", [Online] Available: http://eeweb.poly.edu/~yao/EE4414/memon_F05_v2.pdf
9. Katzenbeisser, Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking".
10. Brian Jacokes, YuliyaKodysh, Andrew Lisy. "Audio Steganography".
11. Robert Krenn, "Steganography and Steganalysis".
12. David Kahn, "The History of Steganography", 1996.
13. P. Chan, M. Lyu, and R. Chin, "Copyright protection on the web: a hybrid digital video watermarking scheme," 2004.
14. M. Ramalingam, "Stego machine - Video steganography using modified LSB algorithm", 2011.
15. Niel Jonson, "Exploring Steganography: Seeing the Unseen", 1998.
16. Sauvik Bhattacharya, "DCT Based Steganography", https://www.slideshare.net/PrimaLCarnagE/dct-steg-o-group-1
17. Tamanna, AshwaniSethi, "Steganography: A Juxtaposition between LSB DCT, DWT", International Journal of Computer Applications, Volume 126 – No.11, 2015.
18. Wikipedia, [Online] Available: https://en.wikipedia.org/wiki/Metadata
19. Metadata Working Group, "Guidelines for handling image metadata", 2010.
20. Thomas Gloe, Andre Fischer and Matthias Kirchner, "Thomas Gloe, Andre Fischer and Matthias Kirchner", 2014.
21. *MotionElements* (2013), https://www.motionelements.com/blog/articles/what-you-need-to-know-about-the-5-most-common-video-file-formats
22. *AVI (Audio Video Interleaved) - video container and Format*, [Online] Available: http://www.ezr8.com/avi.html#3
23. *Wikipedia*, [Online], Available: https://en.wikipedia.org/wiki/Frame_rate
24. Multimedia functions (windows), http://msdn.microsoft.com/.
25. *Wikipedia*, *Audio Video Interleave*, [Online], Available: https://web.archive.org/web/20170411001412/http://www.alexander-noe.com/video/documentation/avi.pdf

26. Joni Monttinen, “The Security of Advanced Encryption Standard”, 2015.
27. Rajasekaravarman.S, Joshna.S,, “Symmetric Key Algorithms A Comparative Analysis”, International Journal of Innovative Research in Computer and Communication Engineering, Vol.4, 2016.
28. P. Princy, “A Comparison of Symmetric Key Algorithms DES, AES, BLOWFISH, RC4, RC6”, International Journal of Computer Science & Engineering Technology (IJCSET), Vol.6, 2015.
29. *Wikipedia*, [Online] Available: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
30. Doug Carner, “Detect and prevent file tampering in multimedia files”.
31. Michael P. Marcinak and Bijan G. Mobasseri, “Digital video watermarking for metadata embedding in UAV video”, In: Military Communications Conference, IEEE.
32. Melinos Averkiou, “Digital Watermarking”.
33. Huang H-C, Fang W, “Metadata-based image watermarking for copyright protection”, 2010.
34. Masoumia M., Amirib S., “A blind scene-based watermarking for video copyright protection”, AEU Int J Electron Commun 67:528–535, 2013.
35. Ahuja R and Bedi SS, “Copyright protection using blind video watermarking algorithm based onmpeg-2 structure. In: International Conference on Computing, Communication and Automation (ICCCA2015) IEEE, 2015.
36. Mstafa RJ & Elleithy KM, “A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes Multimed Tools”.
37. Liu S, Chen DB-W, Gong L, Ji W and Seo S, “A real-time video watermarking algorithm for authentication of small-business wireless surveillance networks”, International Journal of Distributed Sensor Networks, 2015, [Online] Available: http://dx.doi.org/10.1155/2015/789536
38. Monika R. Chourasiya, Prof. Avinash P. Wadhe, “Implementation of video forensics frame work for video source identification”, Satellite Conference ICSTSD, 2016.
39. Naveen Sharma, Anwer Reyaz J., Balan C., “Video Source Identification”, (IJCSIT) International Journal of Computer Science and Information Technologies, 2016.
40. Yanmei Fang, Ahmet Emir Dirik, Xiaoxi Sun, Nasir Memon, “Source Class Identification for DSLR and Compact Cameras”, IEEE.
41. Atanu Maity, “STEGANOGRAPHY – A new technique to hide information within image file”, 2010.
42. *Superuser (2018),* [Online] Available: https://superuser.com/questions/240604/how-do-i-get-the-meta-data-of-a-video-file
43. *Attribute Changer 9*, Version 9.10b – Released 11 August 2018, [Online] Available: https://www.petges.lu/
44. *MSU Quality Measurement Tool*, [Online] Available: http://www.compression.ru/video/quality_measure/vqmt_download.html
45. Basant Kumar, S. P. Singh, Anand Mohan, Animesh Anand, “Performance of Quality Metrics for Compressed Medical Images Through Mean Opinion Score Prediction”, Journal of Medical Imaging and Health Informatics, Vol. 2, 2012.

Appendix A

Listing A.1: File information extraction

```
Publicstring GetFileInfo(stringVideoFilepath)
        {
stringMetaFileName = @"META_DATA.txt";
if (File.Exists(MetaFileName)) { File.Delete(MetaFileName); }
varfilePath = @"" + VideoFilepath + "";
varffProbe = newFFProbe();
varvideoInfo = ffProbe.GetMediaInfo(filePath);
FileStreamfs       =       newFileStream(MetaFileName,       FileMode.OpenOrCreate,
FileAccess.Write);
FileInfo info = newFileInfo(filePath);
string result = "";
if (File.Exists(MetaFileName))
            {
// Create a file to write to.
using (StreamWritersw = newStreamWriter(fs))
                {
DateTime time = File.GetCreationTime(filePath);
sw.WriteLine("\nCT: {0}", time);
time = File.GetLastWriteTime(filePath);
sw.WriteLine("\nLMT: {0}", time);
sw.WriteLine("\nAuthor: {0}", "C.M. KONDASINGHE");
sw.WriteLine("\nLength: {0}", info.Length);

foreach (var stream invideoInfo.Streams)
                    {
if (stream.CodecType == "video")
                        {
sw.WriteLine("\nFrame size: {0}x{1}", stream.Width, stream.Height);
sw.WriteLine("\nFrame rate: {0:0.##}", stream.FrameRate);

result   =   result   +   string.Format("\nFrame   size:   {0}x{1}",   stream.Width,
stream.Height);
result = result + string.Format("\nFrame rate: {0:0.##}", stream.FrameRate);
                        }
                    }
                }
            }
return result;
        }
```

Listing A.2: Secret message encryption

```
privatestaticstringEncryptText(stringplainText, string _key)
        {
string key = _key;
byte[] initVectorBytes = Encoding.UTF8.GetBytes(initVector);
byte[] plainTextBytes = Encoding.UTF8.GetBytes(plainText);
PasswordDeriveBytes password = newPasswordDeriveBytes(key, null);
byte[] keyBytes = password.GetBytes(keysize / 8);
RijndaelManagedsymmetricKey = newRijndaelManaged();
symmetricKey.Mode = CipherMode.ECB;
ICryptoTransformencryptor          =          symmetricKey.CreateEncryptor(keyBytes,
initVectorBytes);
var encrypted = "";

using (MemoryStreammemoryStream = newMemoryStream())
            {
using   (CryptoStreamcryptoStream   =   newCryptoStream(memoryStream,   encryptor,
CryptoStreamMode.Write))
                {
cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);
cryptoStream.FlushFinalBlock();
byte[] cipherTextBytes = memoryStream.ToArray();
encrypted = Convert.ToBase64String(cipherTextBytes);
                }
            }

return encrypted;
        }
```

Listing A.3: Count Carrier Units

```
publicoverridelongCountUnits(){
                    AviReaderar = newAviReader();
                    ar.Open(carrierFile.SourceFileName);
                    Sizesize = ar.BitmapSize;
                    longcountPixels = size.Width * size.Height * ar.CountFrames;
                    ar.Close();
                    returncountPixels;
             }
```

Listing A.4: Count Usable Carrier units

```
publiclongCountUseableUnits(StreamkeyStream){
                    longkeyPosition = keyStream.Position;
                    longcountUnits = CountUnits();
                    longcountUseableUnits = 0;
                    longunitIndex = 0;
                    byte key;

                    while(true){
                           key = GetKey(keyStream);
                           if(unitIndex + key <countUnits){
                                  unitIndex += key;
                                  countUseableUnits++;
                           }else{
                                  break;
                           }
                    }
                    keyStream.Seek(keyPosition, SeekOrigin.Begin);
                    returncountUseableUnits;
             }
```

Listing A.5: Selection of color component

```
//rotate color components
currentColorComponent      =      (currentColorComponent==2)      ?      0      :
(currentColorComponent+1);
//get value of Red, Green or Blue
colorComponent = GetColorComponent(pixelColor, currentColorComponent);

privatestaticbyteGetColorComponent(ColorpixelColor, intcolorComponent){
                    bytereturnValue = 0;
                    switch(colorComponent){
                           case 0:
                                  returnValue = pixelColor.R;
                                  break;
                           case 1:
                                  returnValue = pixelColor.G;
                                  break;
                           case 2:
                                  returnValue = pixelColor.B;
                                  break;
                    }
                    returnreturnValue;
             }
```

Listing A.6: Replacement of secret message with LSB

```
protectedvoidSetBit(intmessageBitIndex,    bytemessageByte,    intcarrierBitIndex,
refbytecarrierByte){
                    //get one bit of the current message byte...
                    boolmessageBit = ((messageByte& (1 <<messageBitIndex)) > 0);
                    //get one bit of the carrier byte
                    boolcarrierBit = ((carrierByte& (1 <<carrierBitIndex)) > 0);

                    //place   [messageBit]   in   the   corresponding   bit   of
[carrierByte]
                    if(messageBit&& !carrierBit){
                           carrierByte += (byte)(1 <<carrierBitIndex);
                    }elseif( !messageBit&&carrierBit){
                           carrierByte -= (byte)(1 <<carrierBitIndex);
                    }
             }
```

Appendix B – PSNR of sample stego videos

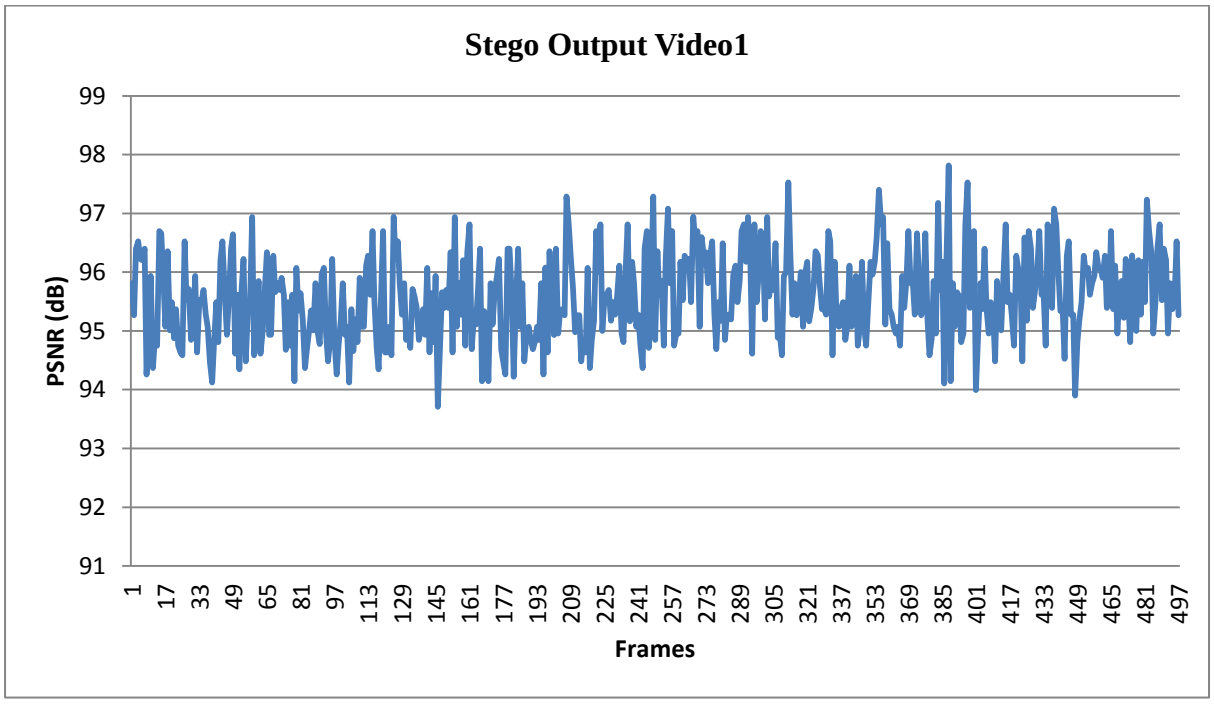

Figure B.1: PSNR of Stego Output Video 1

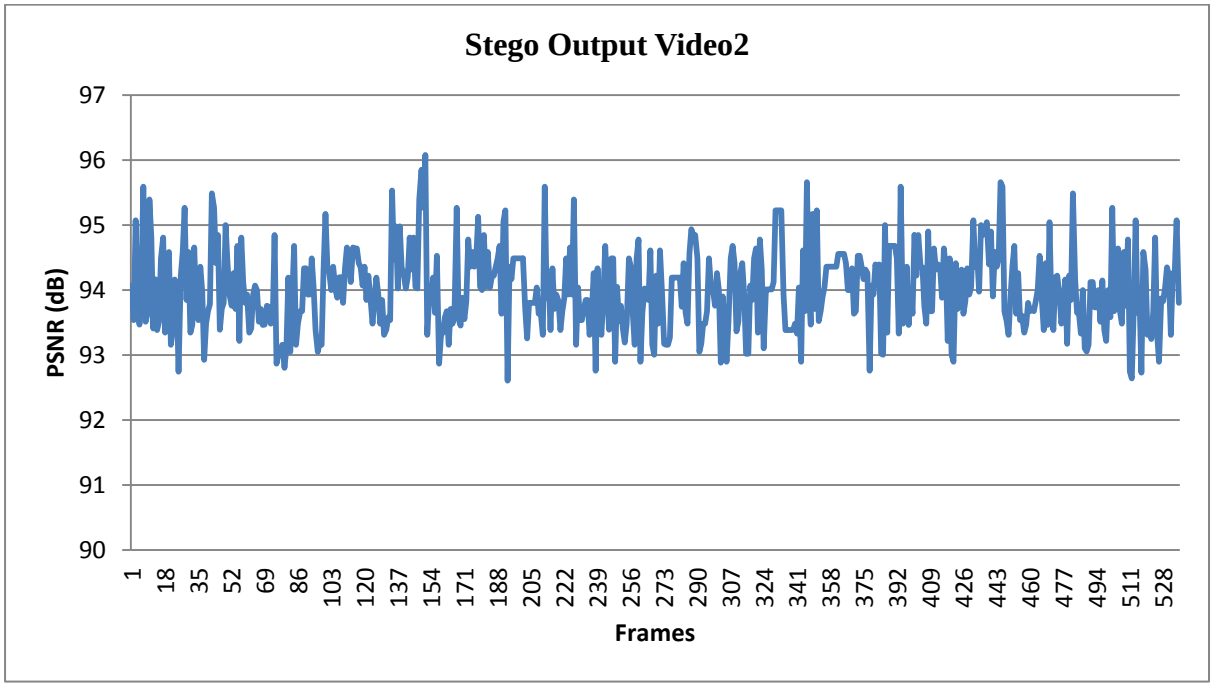

Figure B.2: PSNR of Stego Output Video 2

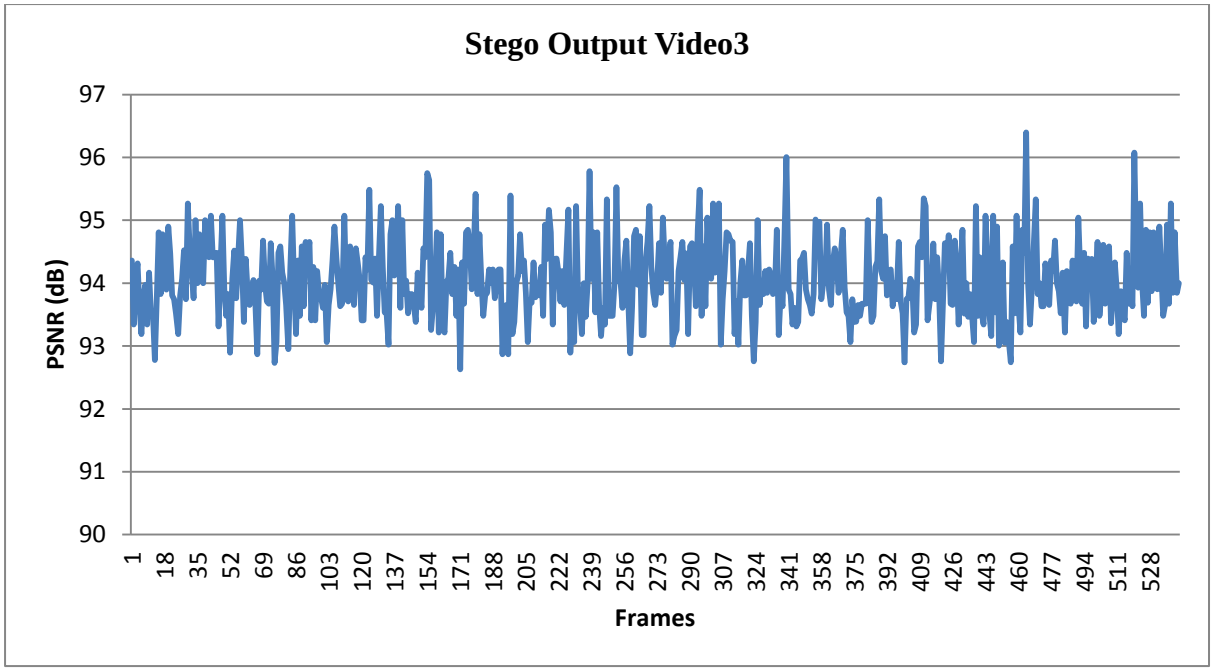

Figure B.3: PSNR of Stego Output Video 3

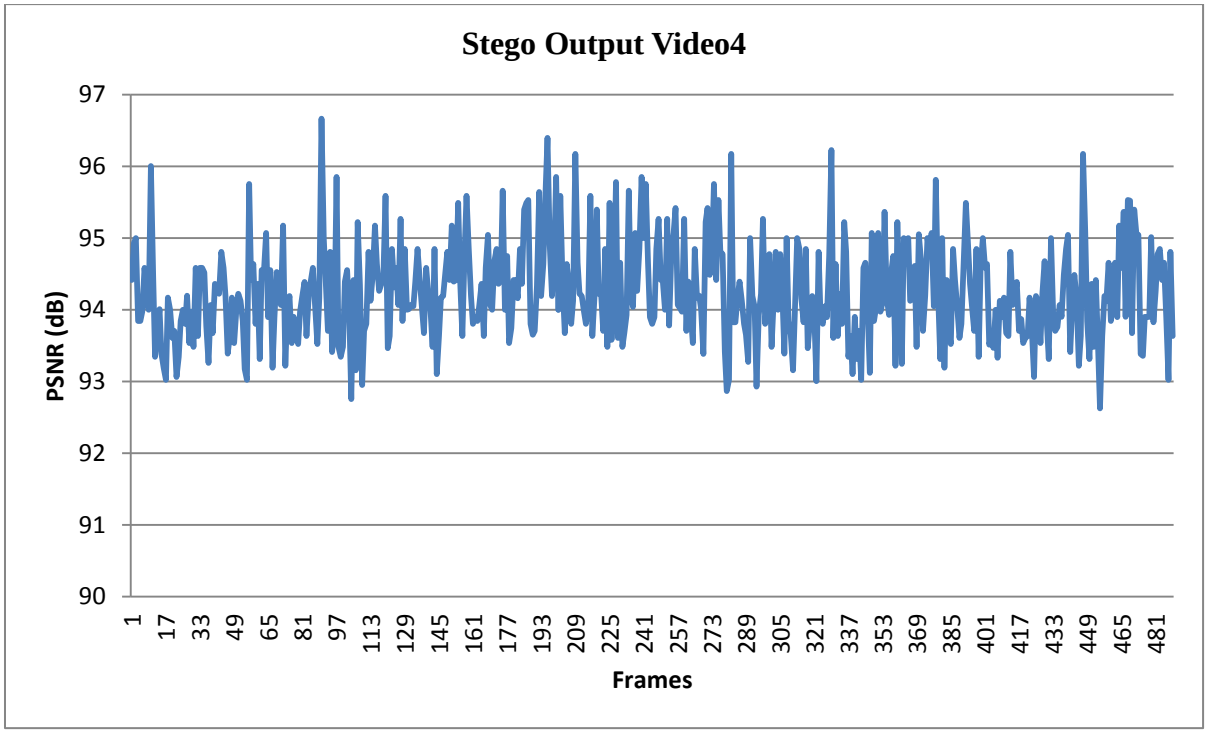

Figure B.4: PSNR of Stego Output Video 4

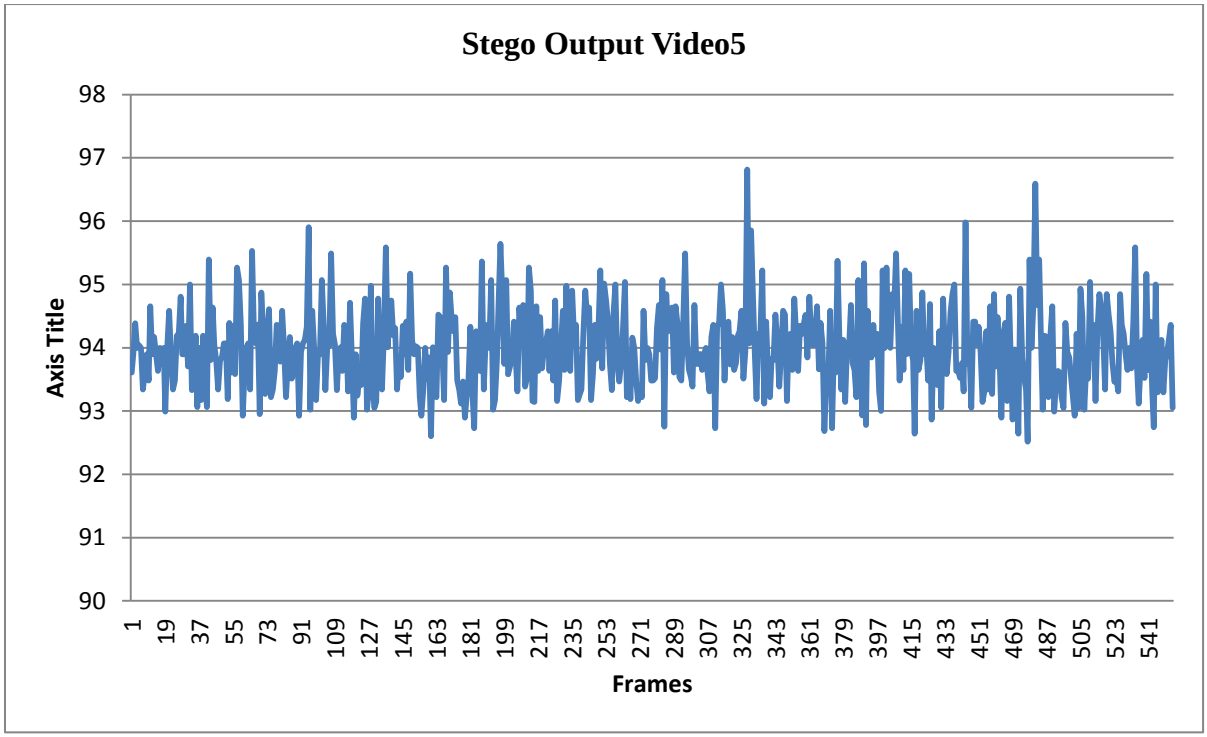

Figure B.5: PSNR of Stego Output Video 5

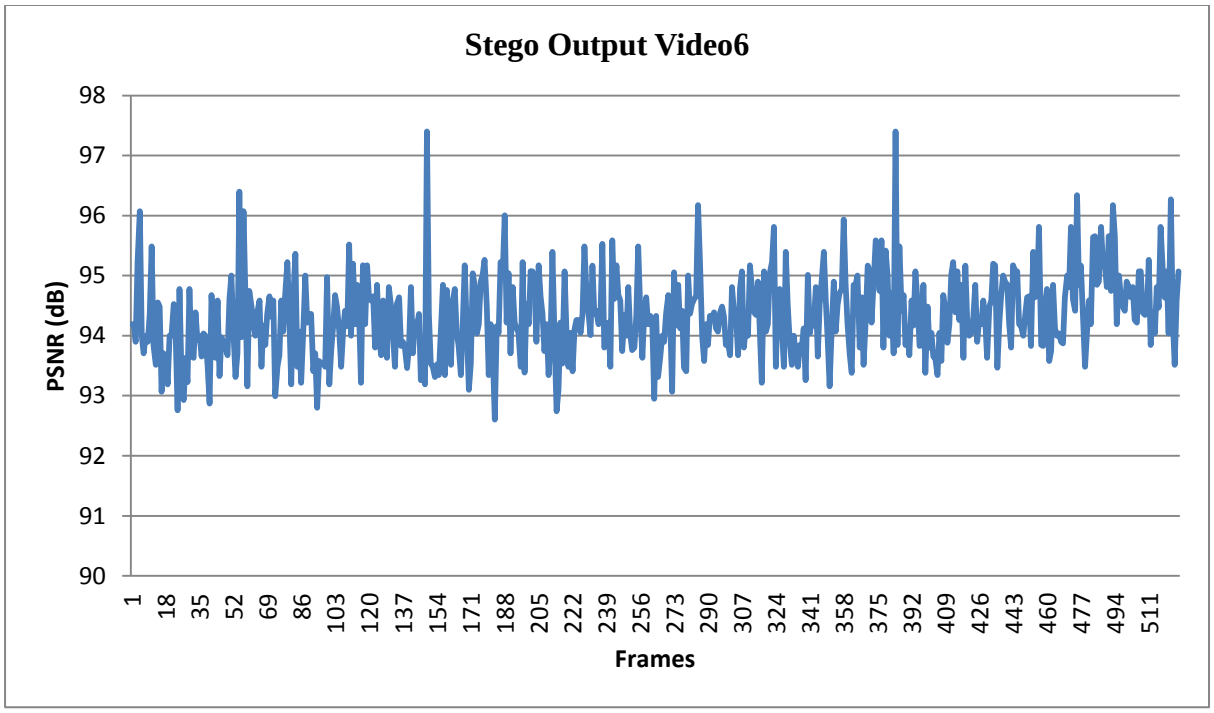

Figure B.6: PSNR of Stego Output Video 6