Ph.D. Thesis

An Improved Video Steganography and Detection of Tampered Frames in a Stego-Video

Submitted by

Asma Sajjad 101-FBAS/PHDCS/F13

Supervisor

Dr. Humaira Ashraf

Co- Supervisor

Dr Nadeem Anjum

Capital University of Science & Technology

Department of Computer Science

Faculty of Computing & Information Technology

International Islamic University, Islamabad.

(2023)

Steganography (Data security)

Video compression

Digital video - Data processing

Image processing - Digital techniques

Computer security

Multimedia systems

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Date: 20/07/23

Final Approval

It is certified that we have read this thesis, entitled "An Improved Video Steganography and Detection of Tampered Frames in a Stego-Video" submitted by Ms. Asma Sajjad, Registration No. 101-FBAS/PHDCS/F13. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the award of the degree of PhD in Computer Science.

Committee

External Examiners:

Dr. Arif Jamal Malik,
Assistant Professor,
Dept.Of SE,Foundation University
DHA, Islamabad.

Dr. Atta Ullah,
Associate Professor of Computer Science,
NUML University,
Islamabad

Internal Examiner:
Dr. Umara Zahid,
Lecturer
Department of Computer Science & Information Technology
FOCS&IT, IIUI

Supervisor:
Dr. Humaira Ashraf,
Assistant Professor
Department of Computer Science & Information Technology
FOCS&IT, IIUI

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Ms Asma Sajjad

Dedication

Dedicated to My Parents and Family

Acknowledgements

First of all, I am very much thankful To Allah Subhanna Wa Taala for Helping me to complete my thesis.

I would like to pay my utmost gratitude to my mother, my husband and my children for their constant prayers, support, time and encouragement that motivated me to complete my research.

Then I am immensely thankful to my Supervisor Dr Humaira Ashraf, Assistant Professor, DCSSE,IIUI for taking me in to supervision after the death of my late supervisor and dear friend, Dr Sadia Arshad, DCSSE, IIUI. May Allah Subhanna Wa Taala Bless her with highest rank in Jannat Ul Firdaus(Aameen).

Dr Humaira Ashraf has been extremely kind and supportive in every step of my thesis. Dear Madam, you have been extremely helpful, kind and a tremendous source of support, and inspiration for me. May Allah Subhanna Wa Taala reward you for your efforts.

Asma Sajjad

ASMA SAJJAD 101-FBAS/PHDCS/F13

Abstract

Utilizing a dual cover medium and a new frame selection algorithm, this study presents an improved method for video steganography. In DNA steganography, a new technique for embedding the stego message into the DNA has been developed. Previously, DNA was utilised as a means of frame selection or as a key in the encryption and decryption process. However, in this study, we employed DNA complementary rules to construct a false DNA from the encrypted message in order to add complexity to the identification process exploiting DNA's cutting properties to embed encrypted data in DNA. This fake DNA is then placed in selected frames of a cover video utilizing a frame selection method including scene change detection. A novel frame selection mechanism for embedding the stego message has been developed, which not only maintains the perceptibility of frames but also increases the security of our embedded message. In this method, cipher text is not introduced to complex frames. These frames are challenging because they involve a scene transition. In this manner, the encryption is concealed behind scene-changing frames. Thus concealing the first possibilities of visual perception. In addition, not even every scene change frame is employed; more security is supplied by selecting a random place inside complex frames using a burger chaotic map. This adds enough complexity to this approach to make it exceedingly challenging for any hacker. After selecting the frame, the fake DNA is implanted in random pixel locations generated using a linear congruence generator, burger chaotic map, and RGB channel. The proposed methodologies yield superior outcomes in terms of MSE, PSNR, and visual perceptibility.

Additionally, the detection of altered frames in steganography videos has been investigated. Previously, stegnographed videos were not investigated for manipulation. In order to receive a complete and recoverable message, however, a steganography movie must also be authenticated. For this premium crop, forgery detection has been implemented in order to spot tampering with counterfeit frames. We have effectively detected both quantitative and qualitative upscale crop and splicing forgeries in steganography video using approaches for identifying such forgeries. Finally, the proposed system has been developed in Matlab in order to evaluate its performance.

TABLE OF CONTENTS:

List of Figures

List of Tables

Chapter 1
Introduction

1. Introduction

Since the time when photography was discovered it is playing a significant part in having an effect on our society's beliefs by constantly altering its perception of facts. Apart from providing entertainment, digital photographs and videos have aided investigations by acting as a repository for evidence suitable for post-incident analysis. Simultaneously, incidences of tampering with video content are increasing. At times, tampered digital content can be practically indistinguishable from real content, having a devastating impact in instances where final choices are made exclusively on the basis of picture and videos. Likewise, a range of other highly sensitive sectors, including law enforcement, politics, and defense might generate "untrustworthy evidence." It is vital to ascertain the authenticity of any digital evidence before to employing it in any judgement.

This chapter provides an introduction of the topics covered in this thesis such as steganography, its history, image and video steganography, cryptography, its history and types are also discussed. DNAs structure and its use in steganography is discussed. This chapter also provides the purpose of this thesis in the form of problem statements and research contributions.

1.1 Steganography

When a secret message is hidden in another file that could be a text file, an image file or even a video file, this is called steganography. The message that a user wants to hide can be in of any form i.e. it could be a text, an image or even a video. The terminology Steganography is composed of two words i.e. “steganos” that means covered, concealed or shielded & "graphein" that means "writing." This term dates back to the year 1499 when Johannes Trithemius coined his famous literal “Steganographia”. It later on masqueraded as a book on magic as it contained a dissertation on cryptography and steganography. Johannes was a German Benedictine abbot, a renowned lexicographer, historian and occultist.

History

The first known examples of steganography are from Greece 440 BC wherein Herodotus recounts several instances in his inspiring work of Western literature. Herodotus was born in Bodrum, Turkey in 424 BC, which was part of the Persian Empire then. Aristagoras He is best known for his compilation of historical booklets on the Greco-Persian Wars. He was also given the title "The Father of History" for his works. Aristagoras was one of his servants. He shaved his head and

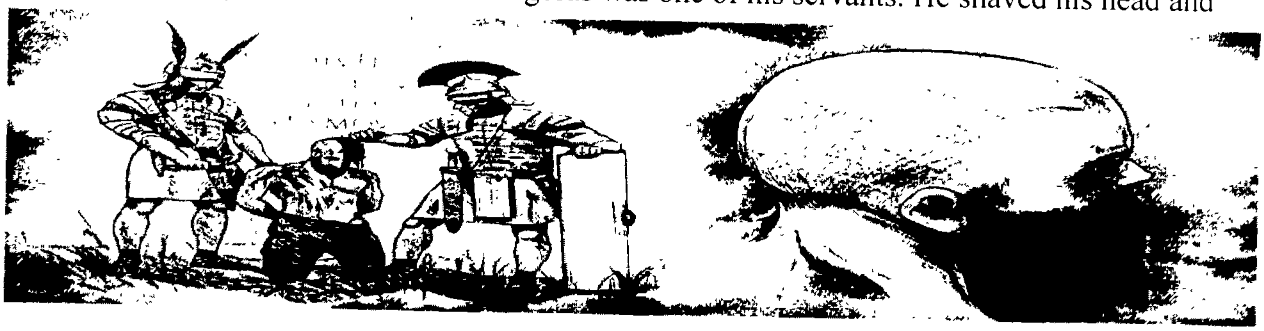


Figure 1.1 Head Shaving for writing the secret message

wrote an admonition on his scalp, "When thou art come to Miletus, bid Aristagoras shave thy head, and look thereon", as illustrated in a shaved and ascribed head in figure 1.1 below.

Egyptians have also used drawings to conceal messages. An illustration of such messages is given in figure 1.2. Hieroglyphics, or sacred carvings, are the name given to ancient Egyptian writing. It is thought to have originated during the Early Dynastic Period (3150-2613 BCE). The ancient Romans also devised invisible inks created with ingredients such as juices of fruits, urine or milk to write invisible text between already written lines of a text.

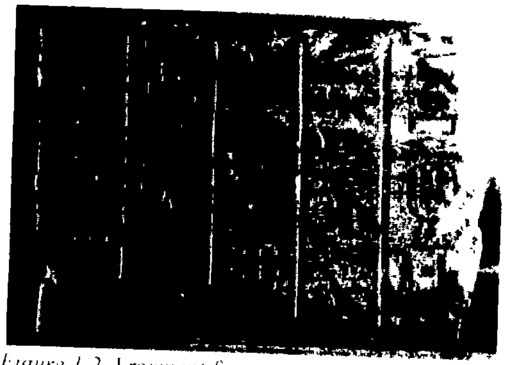


Figure 1.2 Fragment from pyramid of king Pepi I [1]

Steganography in WAR Zones

Different forms of steganography techniques were also used in World War II. One such technique was messages written with invisible inks made from juices of fruits, urine or milk . Gaspari Schotti is also credited with writing the earliest books on steganography titled "Schola Steganographica". It was a 400 page book published in 1665 inspired by the work of Johannes Trithemius works on steganography and cryptography. This followed a series of other publications as fulfilling the demand of those times of war. Another publication was by Auguste Kerckhoff titled Cryptographie Militaire in 1883. After this it was Charles Briquet's work on watermarks in 1907 titled Les Filigraines.

Steganography techniques evolved heavily in the 20th century. Lord Robert Baden-Powell who created the Boy Scout movement was assigned the task of mapping the Boer artillery placements during the Boer War (1899-1902). He disguised his maps in the form of paintings. Those paintings used special patterns on the butterflies' wings to mark the positions of hostile sites. During the early years of the World War II, along with invisible inks, null cyphers were also used. It was plain text mixed with secret message creating up a harmless message about mundane events. Such a message was unlikely to arouse suspicion and thus less likely to be intercepted.
Concluding, the technique of steganography needs a cover medium, a secret message and any steganography function and its converse. It may or may not include a steganography key or password for concealing and recovering the hidden data.

1.2 Steganography Techniques

Steganography is the use of a cover medium to convey a message. A plaintext, still imager, audio, video, or even an IP datagram can be used as the cover medium. Simple text Steganography hides the message within plain text through the use of a variety of techniques, including the addition of symbols, special letters, or spaces in the cover medium. Concealing a message in still images is a popular technique that uses the optical abilities used by humans i.e. HVS Human Visual System (HVS) to conceal messages. Modification of LSB i.e. the Least Significant Bit is a mostly used technique for digital images steganography. The LSB technique effectively conceals the binary counterpart of the secret message within the LSBs of each pixel. This technique is quite versatile. This approach is particularly open to attack because of its simplicity. DCT i.e. the discrete cosine transform is also popular to change the image pixel's least significant bit in order to create a compressed steganography image. Audio steganography, a technique to insert a secret message in a digitized audio signal, slightly altering the byte sequence of the accompanying audio file. Other techniques for this type of steganography are phase encoding, echo concealment, least significant bit encoding, and Internet protocol (IP) datagram steganography and spread spectrum encoding steganography.

1.3 Steganalysis

Steganography is identified by steganalysis, a process that analyses several properties of a surveillance medium. The purpose of steganalysis is to decipher the cover medium's concealed message. There are different kinds of steganography discussed below:-

1.4 Kinds of Steganography

I. Text Steganography is a technique for concealing data within text. This obscures the secret data in a text message after every nth character of each word. Numerous techniques exist for concealing data in a text namely Format-Based Approach, the Random and Statistical Approach and the Linguistics Approach.

II. In image steganography an image is used as a cover medium. The data is hidden in the pixel intensities. Due to the large amount of bits in an image it becomes a common source of cover in digital steganography. Images are available in a number of formats hence a large number of algorithms are also available for concealing data within them. Some of the famous techniques of image steganography are the insertion of the least significant bit, encrypting and scattering, encoding redundant patterns, masking and filtering, coding and cosine transformation, and so on.

III. Audio Steganography technique involves concealing information within audio files. This technique masks data contained in WAV files, AU files, and MP3 audio files. Steganography of audio can be accomplished in a variety of ways such as Low-bit encoding, phase encoding, spread spectrum encoding, and parity encoding are all examples of low-bit encoding.

IV. In video steganography a file is hidden within the frames of a video. For this the common approach is that the discrete cosine transform (DCT) modifies the pixel values to conceal the data with in different frames in a video making it undetectable. Video steganography makes use of the H.264, MP4, MPEG, and AVI file formats. LSB technique i.e. the Least Significant Bit Insertion is also one of the popular approaches in this field.

V. Protocol or Network Steganography conceals information by using a network protocol as a cover object, such as TCP, UDP, ICMP, or IP. The OSI layer network model includes concealed channels that may be used for steganography.[2].

1.5 Research Objective

In DNA steganography, the digital information is first encoded into a sequence of DNA bases (A, C, G, T), and then this encoded DNA sequence is embedded into a cover video. The processing and analysis of DNA data is already a common and well-established practice in the fields of biology, genetics and now steganography as well, making it difficult for any unauthorized individuals to detect the hidden information.

The first objective of our research was to utilize DNA itself as a means of hiding secret information rather than as a key for encryption. After achieving an improved embedding algorithm we moved on to searching a complex frame set for embedding. Once embedding was done on selected complex frames, we further moved to find can a tampered frame be detected. We finally concluded that this approach holds great potential for the development of new and secure methods for data hiding and communication in the future.

1.6 Research Contribution

The main contributions of this research are enumerated below:

1. A new technique has been introduced in DNA in steganography for embedding the stego message within the DNA. Previously, DNA was used as a means of frame selection or as a key in the process of encryption and decryption but in this research we have used DNA complementary rules to create a fake DNA out of the encrypted message in order to add complexity to the process of identification.
2. Further, a novel frame selection mechanism has been introduced for embedding the stego message that not only maintains the perceptibility of frames but adds further security to our embedded message. In this technique encrypted message is added in complex frames. These complex frames are those that have scene change involved. In this way the encryption is camouflaged behind frames that involve scene change. Thus hiding the first chances of visual perception. Further not even all scene change frames are utilized. further security is added by choosing random location in complex frames using burger chaotic map. This way enough complexity is added in this scheme that makes it quite difficult for any hacker. The results of the proposed techniques are better both in terms of MSE and PSNR and visual perceptibility. The MSE values of the proposed mechanism in comparison with algorithm [14] are very much improved where as they have slight improvement in comparison with algorithm [2]. The PSNR values of the proposed mechanism are higher in comparison with algorithm [14], algorithm [5] and algorithm [2].

1.7 Thesis Organization

The following is a summary of the remaining chapters in the doctoral dissertation:

- **Chapter 2:** This chapter discusses the research in the field of video steganography focusing on DNAs use in video steganography in various fields. The methods and techniques being used for frame selection have also been discussed here. It also discusses the research in the field of frame tampering in videos.
- **Chapter 3:** This chapter discusses the proposed solution for an improved Video steganography using DNA and intelligent frame selection technique based on detection of frames involving scene change.
- **Chapter 4:** In this chapter we discuss the strategy for detecting tampered frames in a stego video.
- **Chapter 5:** This chapter gives conclusion and presents some future directions of our research.

Chapter 2
Literature Survey

2. Introduction

A literature review is an in-depth examination of prior research on a certain topic. By mentioning past work on the subject, the author is implying that they have read, analyzed, and incorporated that work into their current work. It provides a "landscape" for the reader, allowing him or her to obtain a complete understanding of the field's advancements. This landscape reveals to the reader that the author has absorbed the lion's share of significant prior works on the subject into her or his research. This chapter discusses the research papers on video steganography, frame selection and tempered frames detection in videos. Then the same literature review is given in tabular form.

2.1 Steganography Techniques

In order to fully understand steganography and its implementation, one must be aware of the various sorts of steganography techniques. Figure 2.1 illustrates various steganography techniques that are also discussed below:

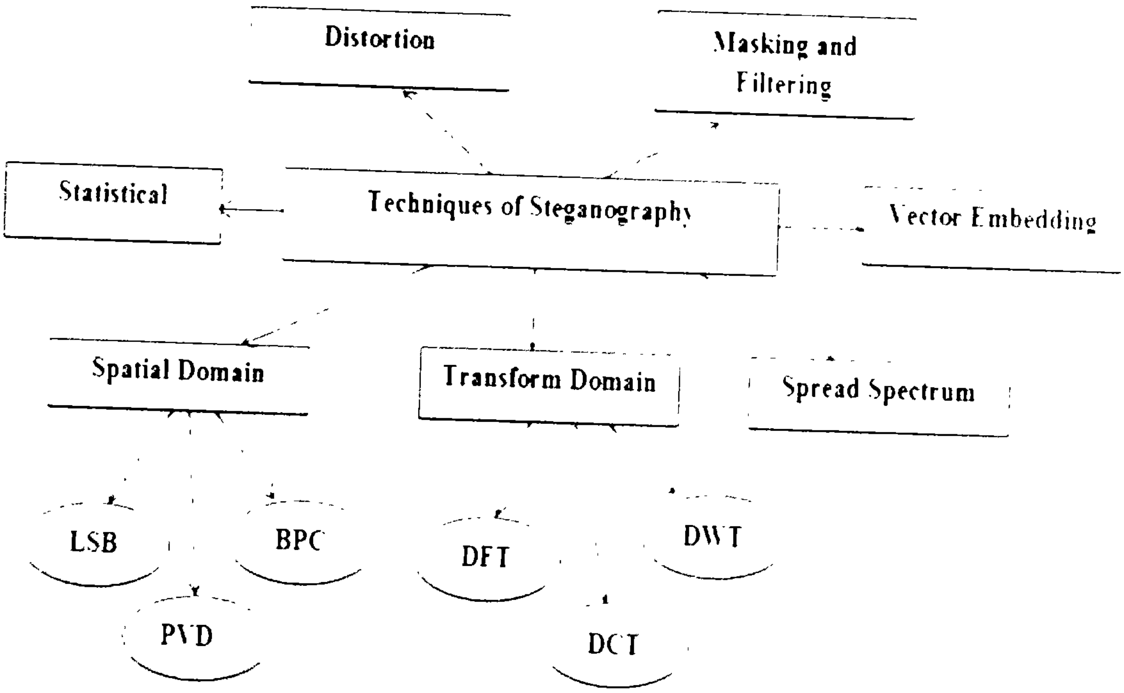


Figure 2.1 Techniques of Steganography

2.1.1 Spatial Domain Methods

In spatial domain the cover image pixels are directly manipulated for masking the secret message. The following are the categories of spatial domain methods:

- **Least Significant Bit LSB:** In this technique certain bits are altered in such a way that human eye fails to detect such a change. Figure 2.2 illustrates an LSB conversion.

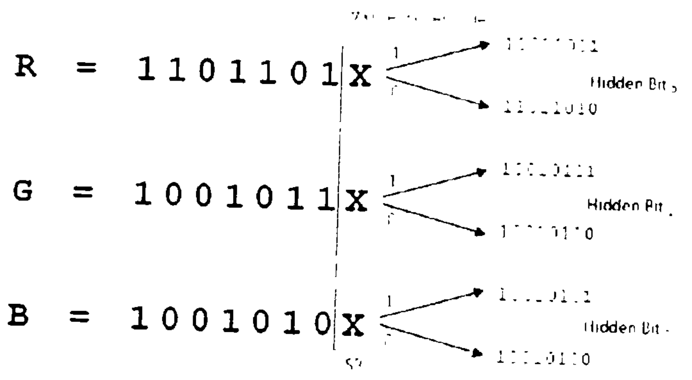
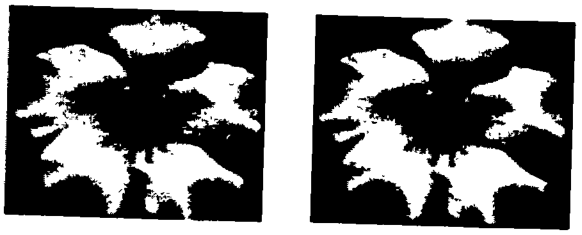


Figure 2.2 LSB Conversion

The LSB approach not only minimizes image degradation but also provides more space for information to be stored in an image in the form of its large no of pixels. Figure 2.3 illustrates an encrypted image using LSB technique. Although it a simple approach but as it undergoes direct pixel manipulation hence it undergoes loss of actual data in the image.



(a) (b)

Figure 2.3 Conversion using LSB (a) original image (b) stego image

- **Difference of Pixel Values:** This technique uses the difference between two consecutive pixels in a block to compute the expected payload.
- **Binary Pattern complexity:** This algorithm determines the noise component in an image complexity and substitutes it with a binary pattern derived from the secret data.

2.1.2 Steganography in Transform Domain

In this domain that data in an image is transformed to another domain before being embedding. It is a more sophisticated technique for concealing information within an image. Numerous methods and manipulations are employed to conceal information in photographs. Certain techniques in this domain are format-independent and can convert between lossless and lossy image formats. Few of its approaches are discussed below:-

- **Discrete Fourier Transform**: The discrete Fourier transform is a completely discrete operation that converts discrete-time signals to a discrete number of frequencies. DFT translates a finite list of equally spaced function samples to a list of coefficients for a finite combination of complex sinusoids in the order of their frequencies. The sampled function is said to be transferred to the frequency domain from its original domain of frequently time or position along a line. The Discrete Time Fourier transform operates in discrete time but translates to continuous frequency via the Fourier transform. The algorithm for computing the DFT is incredibly quick on modern computers. This procedure is called the Fast Fourier Transform, or FFT, and it produces the same result as the DFT by utilizing the Inverse Discrete Fourier Transform.

- **Discrete Cosine Transform:** This method is similar to the Discrete Fourier Transform. DCT transforms the spatial domain of a signal or image to the frequency domain. The mathematical transforms distribute the position of the pixel values throughout a region of the image. Because the image is divided into 8X8 pixel blocks and these pixel blocks are translated into 64 DCT. Each block is subjected to the DCT from left to right, up to down. Each block is compressed using a quantization table to scale the DCT coefficients, and the message is stored in the DCT coefficients. The image's array of compressed blocks is saved in a substantially lesser amount of space. When desired, the image is reconstructed via decompression, which is accomplished by the use of the Inverse discrete cosine transform, abbreviated IDCT.

- **Discrete Wavelet Transform:** It is used to convert a spatial image to a frequency image. During the steganography process, DWT recognizes the high and low frequency information contained in each pixel of the image. It is a mathematical technique for dissecting an image hierarchically. It is mostly used to process signals that are not stationary. The wavelet transform is based on wavelets, which are tiny waves with changing frequency and duration. It specifies both the frequency of the picture and its geographical placement. Mother wavelets are wavelets that are generated by the translation and dilation of a fixed function. DWT is capable of operating in both one- and two-dimensional planes. The DWT is a more accurate multi-resolution image description model than the DFT or DCT. The current standard for picture compression, JPEG 2000, is based on wavelet transforms.

- **Embedding in Vector:**

  This method makes use of a resilient algorithm and codec standards. This approach adds sound into the video frames. It uses the H.264 coding standard. The inserted data has no influence on the visual or statistical invisibility of the movie sequence.

- **Spectrum Spreading:**

This approach spreads the secret data across a wide frequency spectrum. The lower a frequency bands signal intensity to noise intensity the more difficult it becomes to identify the presence secret data. Very large amounts of information can be concealed throughout the available spectrum giving the advantage that even if few bands of data are identified, there will still be sufficient information in the remaining bands to be recovered. As a result, erasing the data fully without ruining the cover is challenging. It is a highly effective method of military communication.

- **Statistical Technique:**

Messages are incorporated in this process by altering many attributes of the cover medium. For this the cover medium is first segmented in to blocks. One bit is embedded in each block.

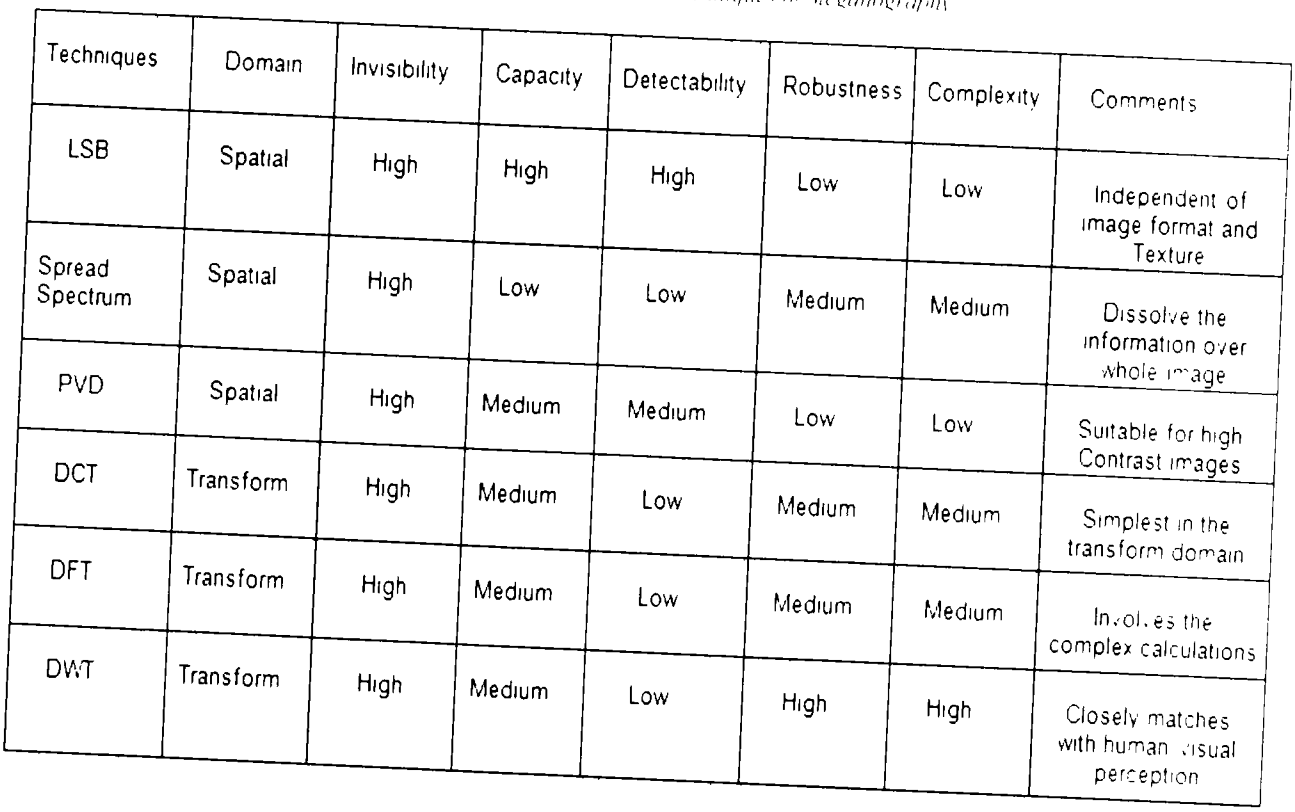
- **Distortion Techniques:**

As the name states, in this technique the cover medium is distorted before storing the secret data i.e. the cover image is altered before embedding. Decoding of data is done using some secret key.

- **Masking and Filtering:**

This technique uses a watermark to disguise the data. This technique is useful when watermarks are used in an image because data is then concealed in the watermark of the image. This technique can be used without destroying the image. Grayscale and 24-bit images are created using this technique.

Table 2.1 Comparison of various techniques of Steganography



Techniques	Domain	Invisibility	Capacity	Detectability	Robustness	Complexity	Comments
LSB	Spatial	High	High	High	Low	Low	Independent of image format and Texture
Spread Spectrum	Spatial	High	Low	Low	Medium	Medium	Dissolve the information over whole image
PVD	Spatial	High	Medium	Medium	Low	Low	Suitable for high Contrast images
DCT	Transform	High	Medium	Low	Medium	Medium	Simplest in the transform domain
DFT	Transform	High	Medium	Low	Medium	Medium	Involves the complex calculations
DWT	Transform	High	Medium	Low	High	High	Closely matches with human visual perception

2.2 Steganography Metrics

The efficiency of the steganography approach is assessed by well-known measures such as robustness, imperceptibility, bit error rate, means error and peak signal to noise ratio. Each of these measures are described below:

- **Robustness:** It is the ability of embedded data to keep its integrity when subjected to various types of transformations.
- **Imperceptibility:** Steganography's power is in its capacity to remain undetectable to the naked eye, this is the first and most critical prerequisite.
- **Bit Error Rate:** It is the number of bit errors per unit time. In case of image it is the no of bits altered due to any kind of alteration, distortion or damage during transmission.
- **Mean Square Error:** Mean square error is calculated by comparison of two images byte by byte. MSE can be used to calculate image distortion using an 8-bit pixel representation or a grey-level image representation with up to 256 levels.
- **Peak Signal to Noise Ratio:** Any steganography approach must concealed secret data within the image without degrading the image's quality. PSNR is used for evaluating the quality of the reconstructed image or video subject to lossy compression. A higher PSNR number indicates a higher image quality, i.e. less error.

All of the above strategies are capable of safeguarding the secret data. Certain techniques have a high temporal complexity but store very little info in the images. Steganography algorithms are a rapidly growing field of research in which new and more efficient algorithms are emerging every day.

2.3 Steganography Attacks/ Stego Analysis Methods

Steganography attacks involve the detection and extraction of hidden artefacts within and on the steganography media, followed by tampering with or destruction of the steganography media. Steganography attacks are followed by stego analysis in order for the intruder to detect and recover the hidden information. Steganographic attacks operate based on the type of information available for stego analysis. This data may contain the concealed message, carrier (cover) medium, stego-object, steganography tools, or data concealment techniques. As a result, stego analysis is classified into six categories: stego-only analysis, known-stego analysis, known-message analysis, known-cover analysis, chosen-message analysis, and chosen-stego analysis. Several well-known attacks include the following:

- **Stego-only attack:** In a stego-only attack, the steg analyst or attacker has no access to any information other than the stego-medium or stego-object. To retrieve the information concealed in this attack, the stag analyst must attempt every steganography algorithm and associated attack.
- **Known-stego attack:** The attacker gains knowledge of the steganographic algorithm as well as the original and stego-objects through this attack. With this knowledge in hand, the attacker can deduce the information that is hidden.

- **Known-message attack:** The known-message attack requires the presence of both the message and the stego-medium. This attack can be used to ascertain the method employed to conceal the message.

- **Known-cover attack:** The known-cover attack is used when attackers are aware of both the stego-object and the initial cover-medium. This enables a comparison of the two mediums in order to detect changes in the medium's format and therefore to deduce the secret message.

- **Chosen-message attack:** To determine the steganography algorithm used to conceal the information, a stego-object is generated to identify patterns that might suggest the use of particular steganography tools or method to identify the steganography technique used.

- **Chosen-stego attack:** When both the stego-object and the steganographic instrument or method used to conceal the message, the chosen-stego assault occurs.

2.4 Detecting Steganography

An investigator's consideration of steganography may be prompted by previously examined crimes. Child pornographers employ steganography to disguise sexual information that is placed on a website or transmitted via email. Additionally, steganography is employed in business-related crimes for concealing critical records for crimes such as terrorism, theft and smuggling. Therefore. an investigator should start looking for steganographic clues in files and papers. They should be familiar with common steganographic tools, software, and websites. This knowledge will help them locate steganography algorithms and software's. They can start by checking the suspect's file names and online pages by searching in the browser history, browser's cookies, and even the key registry entries. This material is vital to the investigation and guides the investigator in later steps.

Other programming files must be examined for information about the hidden files. Investigators should look into the binary editors, disc wiping softwares, and chat softwares that are used to transform data from one code to another also.

- **Multimedia Files:** Investigators should do a system search for huge files that could be used as cover medium. Usually computers have several small sized audio and image files but if a high volume of huge files, is found, investigators may suspect that these files are enormous carrier files. This is conceivable if your computer system has a high amount of duplicate files.

- **Text Files:** Many times characters in text files are altered to conceal the data. Text patterns or disruptions, the language used, line height, and an abnormally large number of blank spaces can all be used to detect alterations in text files.

- **Image Files:** To determine steganography in an image, investigators may examine the file's size and format, latest modified time stamp, color palette, any distortions in images making the image blur, any abnormalities in terms of color composition, brightness, pixel relationships, and so on.

- **Audio File:** This conceals personal information such as private documents and files within audio files. Such techniques use a range of implementation techniques, bandwidths, and mechanisms for disguising conventional data.

- **Video File:** Secret data is stored in video frames in this technique. This data is decrypted using a number of algorithms similar to those used for decrypting image and audio files. Majority of methods for detecting steganography in videos involve human interaction due to the presence of large variations.

2.5 Steganography Detection Tools

There are many steganography detection tools available few are listed below [3].

- **Xstegsecret:** Xstegsecret is a Java based steganalysis tool that analyses digital material for hidden info and detection of FOF, LSB, and DCTs.

- **StegSecret:** Stegsecret is a cross-platform freely available steganalysis tool based on Lava for detecting concealed data in various types of media. It utilizes EOF, LSB, DCTs, and other techniques to uncover hidden data.

- **StegAlyzerAS:** This tool locates entries in file and entries in registry associated with software used for steganography. It uses the Steganography Application Fingerprint Database's SAFD, CRC-321 MD5 and SHA-512 hash values to identify files.

- **StegAiyzerRTS:** It is a real-time security system for networks that identifies and blocks the use of digital steganographic programs. It identifies steganographic programs downloads by comparing fingerprints to a database of over 960 steganographic apps.

- **StegExpose:** StegExpose is a steganography detecting tool for lossless PNG and BMP images. It uses a command line interface to analyst photos in bulk and provides reporting and customizing possibilities for non-forensic professionals.
- **StegAiyzerS5:** StegAlyzerSS is a steganography program that enables examiners to identify approximately 55 signatures left by steganography apps inside the files while injecting secret information.
- **Steganography Studio:** The Steganography Studio software teaches users how to use certain highly configurable algorithms and filters. Additionally, it employs the most powerful image analysis techniques to unearth hidden data.
- **Virtual Steganographic Laboratory (VSL):** The Virtual Steganographic Laboratory (VSL) tool assists in the concealment of large amounts of data in digital images. It is also capable of detection of the concealed data, and provides tools for testing many algorithms for steganography.
- **Stegdetect:** Stegdetect automatically finds steganographic information in a collection of images. It is capable of detecting a wide number of steganographic techniques used to conceal information in JPEG images.
- **ImgStegano:** Stegcletect is a tool that automatically finds steganographic information in photographs. ImgStegano assists in identifying steganography on bmp or png images by detecting a variety of techniques used for steganography. It identifies image steganography through the use of an enhanced LSB technique.
- **Gargoyle Investigator ™Forensic Pro:** It is an application that conducts quick searches for known malicious programs and contraband over a network and even within archived files. It can easily detect botnets, any key loggers also. This software is also capable of identifying stego files created by tools such as 5-Tools, Blindside. Its datasets including almost 20,000 distinct types of harmful applications and softwares. At the end it gives a detailed report in xml format along with source time stamp information.

2.6 Cryptography

To turn a plaintext into an encrypted document known as the cypher text, cryptography employs an algorithm and a key. A safe algorithm is one that is complicated enough that an attacker cannot deduce any attributes of the plaintext or key from the cypher text. If the same key is utilized, a given algorithm will always turn the same plaintext into the same cypher text. Cryptography can be symmetric and asymmetric.

- **Symmetric Vs Asymmetric cryptography?**

Symmetric cryptography uses same key for encrypting data as well for the data decryption. Before key transmission, sender and recipient must communicate it to ensure that both parties are aware of it. Asymmetric cryptography employs distinct keys for its encryption and decryption procedures. Each user is endowed with two keys one is public and the other is private. While the private key is kept private, the public key is freely shared. Only the private key linked with the public key is capable of decrypting data encrypted using the public key. Only the public key that accompanies the private key is capable of decrypting data encrypted with the private key. Asymmetric cryptography is slow and can encrypt only data less than the key size, but symmetric is fast and best for converting large volumes of data.

Cryptography is capable of encrypting communications transmitted through unsecure networks. The two fundamental types of network assaults are passive and aggressive attacks. Passive attacks involve an attacker probing a network segment for delicate information, such as online hacking and decrypts hacked data later where as in an active attack the attacker modifies the network traffic online. Cryptographic protocols safeguard data against hostile snooping and manipulation.

- **Symmetric Ciphers**

Such cyphers utilize same key for encrypting data as well for the data decryption. Recipient and the sender therefore, come to an understanding on a private key in advance. A symmetric algorithm's cryptographic strength can be determined by key size it uses. Advance Encryption Standard (AES), Blowfish and Data Encryption Standard (DES), and widely used symmetric cyphers. The DES algorithm employs a 64-bit key. In lieu of DES, Triple DES can be used to protect data. Blowfish supports key lengths ranging from 32 to 448 bits, but the majority of implementations use 128-bit keys. In 2002, the US government switched from DES to AES as their encryption standard. AES has gained popularity over the years due to its ability to combine the speed of DES with the security of Triple DES. Numerous public AES-based commodities come standard with 128-bit secret keys; numerous public AES-based goods come standard with 192-bit furtive keys. These algorithms are popular due to their speed, which encrypts massive amounts of data efficiently.

As with all forms of cryptography, symmetric key cryptography begins by encrypting a message. They are capable of encrypting massive amounts of data and are far faster than asymmetric cyphers. They do, however, require sophisticated mechanisms for securely disseminating the secret keys to both parties. Using this encryption procedure, any plaintext data will be turned to cypher text, an unreadable code. The encrypted text is subsequently transmitted to a third party, who decrypts it in order to recover the original message. The sender and recipient ensure beforehand about key sharing as a single key is used at both sender and receiver ends. As a result, the encryption and decryption algorithms have the following structure shown below in figure 2.4:

Encrypt (plaintext, key) =cipher text,
Decrypt (cipher text, key)} = plaintext

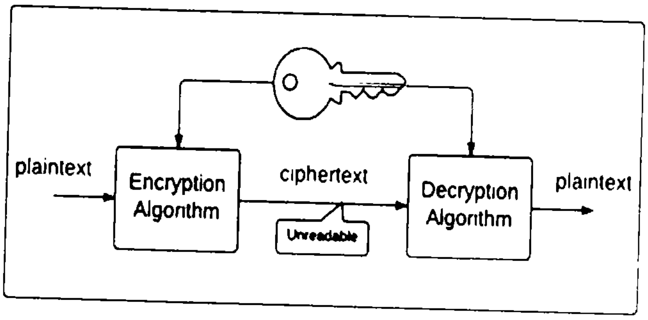


Figure 2.4 The symmetric encryption process

Stream cyphers and block cyphers are the two types of symmetric cyphers.

2.6.1 Stream Symmetric Ciphers

Stream cyphers are designed to generate an infinite number of cryptographic keystreams from random input. They encrypt each bit (or byte) of input data one bit (or byte) at a time. Rather than splitting plaintext material into smaller bits, stream cyphers work with a continuous stream of it.

1 OTP (One Time Pad)

Gilbert Vernam, an engineer with AT&T Corporation in the United States, devised this cypher in 1917. It has been demonstrated that if OTP is utilized correctly, it is impossible to crack. It provides absolute secrecy and enables for lightning-fast encryption and decryption. The secret key, on the other hand, must be at least as long as the message, making it cumbersome to employ for delivering vast amounts of electronic data. The same method is used for data encryption and decryption when utilizing OTP. The message's (or cipher text's) bytes are XORed with the secret key's bytes. One by one, the bytes are added, and each addition results in one output byte as shown in formula below:

$$m_i \text{ XOR } k_i = c_i$$

$$c_i \text{ XOR } k_i = m_i \qquad \text{(i)}$$

Using the same key repeatedly

Only one element of the secret key can be used to encrypt one part of the message at a time (of course, of the same length). The attacker can discover the two original messages summed by XOR by using the same key bytes several times. Using attacks based on language and encoding properties, the intruder can attempt to break the cypher using two original messages averaged by XOR using the equation below where C_1 and C_2 is equal to the $M_1$and M_2 Xored with the key.

$$M_1 \text{ XOR } K = C_1$$

$$M_2 \text{ XOR } K = C_2$$

$$C_1 \text{ XOR } C_2 = M_1 \text{ XOR } K \text{ XOR } M_2 \text{ XOR } K = M_1 \text{ XOR } M_2 \qquad \text{(ii)}$$

Providing no integrity

It is possible to alter the cipher text in such a way that the receiver is unaware of the change. Worse, the alterations have a predictable effect on the message. The attackers can only change the parts of the message that they want if they know the structure of the message. The attacker's change is denoted by the letter p in the formula below.

$$m \rightarrow \mathrm{enc}\,(m, k) \rightarrow m \text{ XOR } k$$

$$(m \text{ XOR } k) \text{ XOR } p = m \text{ XOR } k \text{ XOR } p$$

$$m \text{ XOR } k \text{ XOR } p \rightarrow \mathrm{dec}(m \text{ XOR } p, k) \rightarrow m \text{ XOR } p \qquad \text{(iii)}$$

Secret Sharing

The secret key can be shared among a group of people via OTP. Only when all of those parties use their parts of the key can the encrypted content be deciphered. Only one sub key will be known by each participant. To encrypt a text of size n using a secret key shared by m persons, m*n key characters must be prepared. As a result, each sub key will have n characters and can encrypt a text of up to n characters. If a secret key is shared by three people, for example, all three sub keys must be XORed with the cipher text in order to recover the original message as shown in figure 2.5 below.

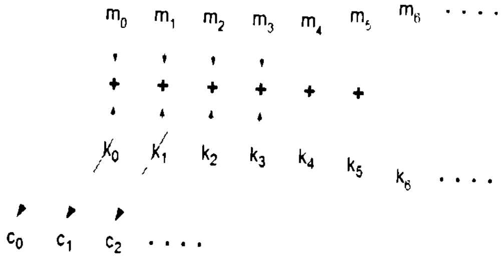


Figure 2.5 Block Diagram of OTP Algorithm

XOR Operation in OTP

Exclusive OR is the only operation performed during OTP encryption and decryption (XOR). One by one, the key bytes are XORed with the data bytes. Each time, the first byte's eight bits are XORed with the second byte's eight bits.

2. RC4

This stream cypher employs a symmetric secret key. The length of the key can be up to 2048 bits. It is well-known for its simplicity and speed. It was founded in 1987 by Ron Rivest of RSA Security. The implementation of the RC4 cypher was unknown until September 1994, when it was anonymously submitted to the Cypherpunks email group. A symmetric secret key is used in this stream cypher. The key's length can be up to 2048 bits. It is well-known for its speed and simplicity of usage. It was founded in 1987 by Ron Rivest, the founder of RSA Security.

Because it only manipulates single bytes, the RC4 algorithm was created specifically for use in software applications. It does not employ LFSR registers, which can be implemented optimally in hardware but are slow in applications, unlike many other stream cyphers. The cypher was developed a long time ago and has various flaws that have since been addressed in contemporary stream cyphers. Keystream byte values that are somewhat more likely to occur than other combinations can be found. Several bytes like that have been discovered in the last 20 years. This flaw has led to the discovery of some attacks.

The insufficient key schedule is probably the most serious flaw in the RC4 cypher. Because to this flaw, the first bytes of the keystream can be used to deduce some information about the secret key. It is recommended that the first few bytes of the keystream be discarded. RC4-dropN is the name given to this enhancement, where N is usually a multiple of 256. For each encryption, RC4 does not require a distinct nonce in addition to the key. As a result, the cryptosystem must account for unique keystream values and describe how the nonce should be combined with the original secret key. The optimal approach would be to hash the nonce and key together to create the RC4 keystream's foundation. Many apps, unfortunately, simply concatenate key and nonce, leaving them vulnerable to so-called related key attacks. This RC4 flaw was used in the Fluhrer, Mantin, and Shamir (FMS) attack on WEP, which was released in 2001. Figure 2.6 illustrates the block diagram of RC4 below.

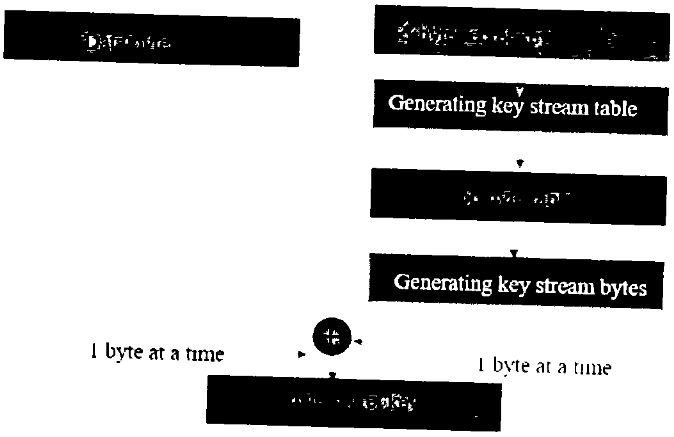


Figure 2.6 Block Diagram of RC4

3. Salsa20

It is a fast and efficient stream symmetric cypher with a key length of 32 bytes and a symmetric secret key. Daniel Bernstein, a research professor of computer science at the University of Illinois at Chicago designed it in 2005. It was submitted to the e STREAM project, which operated from 2004 to 2008 with the goal of fostering research into stream cyphers. There are currently no known or successful attacks against the Salsa20 family of encryption algorithms. Figure 2.7 illustrates the block diagram of Salsa20 below. It operates on 64-byte data blocks using a hash function and returns a 64-byte cypher text block. Decryption uses the same algorithm.

Its other implementations are Salsa20/8 and Salsa20/12, as well as Bernstein's 2008 Cha-cha family of algorithms. They provide significantly more security than the original Salsa20 cypher due to the use of slightly better hash algorithms.

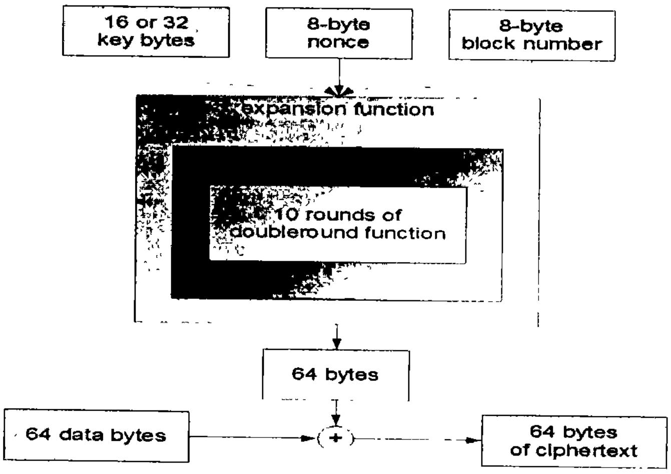


Figure 2.7 Block Diagram of Salsa20 Algorithm

3. CSS (Content Scramble System)

CSS a stream cypher was developed in 1996 by the DVD Forum as a technique of encrypting DVD content. It makes use of a 40-bit key length. Due to the key's poor design, its effective length is approximately 16 bits. In 1999, it was compromised as a result of a brute-force attack.

2.6.2 Asymmetric Ciphers

Asymmetric cyphers avoid the transmission of a shared secret key via an insecure channel, such as the internet. The sender and receiver encrypt the message using the recipient's public key. Only the owner's second secret key may decode the data. In this two keys are used one is called the public key and the other is called the private key. While the public key is freely distributable, the private key must remain private. Text encrypted using private key is decoded using just the associated public key. Symmetric cryptography believed to be optimal and fast for encoding vast quantities of data as whole disc partition or database, whereas asymmetric cryptography is quite slow yet capable of encrypting data less than the key's size. The following section discusses a few asymmetric cyphers.

1 RSA

RSA is an algorithm extensively used for public key encryption. It is capable of encrypting both messages and digital signatures. It uses a key length of approximately 1000–4000 bits. In 1977, Ron Rivest, Adi Shamir, and Leonard Adleman designed RSA. Equation iv illustrates RSA encryption and decryption. A plaintext attack is possible because of its deterministic algorithm. A known public key can be used to encrypt a large number of messages. As a result, an attacker can deduce the contents of seized encrypted messages by comparing them to messages he has constructed. It's utilized in a variety of applications, protocols, and communication methods. The RSA cryptosystem involves a pair of mathematical equations for encryption and decryption:

The RSA cryptosystem is a widely-used asymmetric encryption algorithm that involves a pair of mathematical equations for encryption and decryption. These equations rely on the properties of prime numbers and modular arithmetic to secure communication. Here's a simplified explanation of how RSA encryption and decryption work:

1. **Key Generation:**
   - Choose two large prime numbers, p and q.
   - Calculate their product, n = p * q. This is the modulus for both the public and private keys.
   - Compute Euler's totient function φ(n) = (p - 1) * (q - 1). This is used to calculate the private exponent.
   - Choose a public exponent e, typically a small prime number such as 65537, that is coprime to φ(n). This forms the public key (e, n).
   - Calculate the private exponent d such that (d * e) % φ(n) = 1. This forms the private key (d, n).

2. **Encryption (using the public key):**
   - Convert the plaintext message into an integer M.
   - Compute the ciphertext C = M^e mod n.

3. **Decryption (using the private key):**
   - Receive the ciphertext C.
   - Compute the plaintext message M = C^d mod n.

The security of RSA relies on the difficulty of factoring the modulus n (the product of two large prime numbers) to derive the private key. As long as this factoring problem remains computationally hard, RSA is considered secure. These equations ensure that data encrypted with the public key can only be decrypted with the corresponding private key, providing a secure means of communication and data protection.

2 Diffie–Hellman Protocol

Two American cryptographers, Whitfield Diffie and Martin Hellman, released the algorithm for the first time in 1976. It is considered to be a little faster than RSA.

This algorithm generates a shared secret key. The key can then be used to secure future communications using symmetric encryption. Any eavesdropper will be unable to deduce the key's location. According to mathematical theory, the technique is based on discrete logarithms in given groups. Asymmetric data encryption is possible using this technology. Man-in-the-middle attacks are possible since the protocol lacks authentication. Figure 2.8 illustrates the Diffie-Hellman protocol exchange scenario.

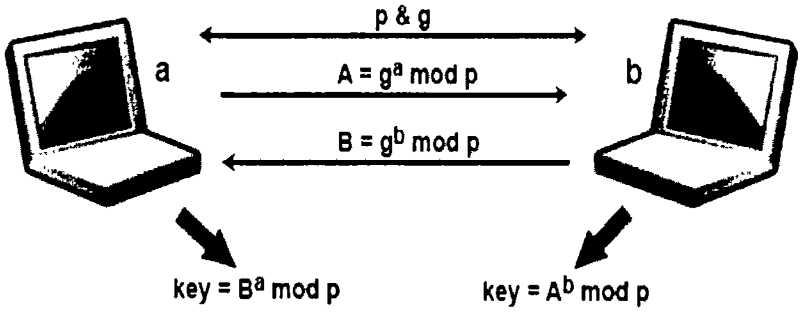


Figure 2.8 Diffie-Hellman protocol exchange scenario

3 AES

AES is widely considered to be the best symmetric key encryption algorithm "It is considered the "gold standard" for data encryption. The AES algorithm is NIST-certified and is used by the US government to protect sensitive data." AES has become the de facto standard symmetric key cypher for virtually everyone due to its ability to "guard" data. It is an open standard that may be used for any purpose, whether public or private, commercial or non-commercial. AES is a cypher for symmetric key encryption. This implies that the data is decrypted using the same key that was used to encrypt it in the first place. It can only be decrypted by the intended recipient who owns the corresponding private key.

AES is commonly recognised as the world's most secure symmetric key encryption technology. Other symmetric key cyphers, such as Twofish, which was designed in collaboration with renowned cryptographer Bruce Schneier, are likewise extremely secure. However, these cyphers have not been subjected to the same rigorous testing as AES. When done correctly, AES is unbreakable. In 2011, the Fujitsu K estimated, brute-forcing a 128-bit AES key would take Fujitsu K about 1.02 x 10^{18} years - or nearly one billion billion (one quintillion) years. In 2017, China's Sunway TaihuLight supercomputer brute-forcing a 128-bit AES key would take 885 quadrillion years even on the world's fastest computer. To brute force a 256-bit cypher, 3.31 x 10^{56} operations are required. This is roughly equal to the total number of atoms in the universe!

Cryptography specialists identified a hole in AES in 2011 that enabled them to break it four times faster than ever before. "To put this in context," one of the researchers explained at the time, "it would take almost two billion years to recover an AES-128 key on a trillion machines capable of evaluating a billion keys per second." In response to this exploit, AES-128 encryption was extended by four rounds (see below) to strengthen its security margin.

The key to AES encryption is only secure to the extent that the encryption is secure. These keys are virtually usually protected by passwords, and we are all well aware of our inability to create strong passwords. Passwords that safeguard AES keys can also be compromised via virus-infected key recorders, social engineering attacks, and other ways. Password managers, as well as two-way firewalls, effective antivirus software, and enhanced security awareness, all help to mitigate this issue significantly. This issue is handled by encrypting data using a shared public key by AES.

2.7 Videos

The difference between a picture and a video is that a photograph is a single image. but a video is a sequence of images that are played back at a predetermined frame rate (analog). The electronic depiction of video in the form of encoded digital data is known as digital video. This is in contrast to analogue video, which uses analogue impulses to describe moving visual images.

The First Ever Motion Picture

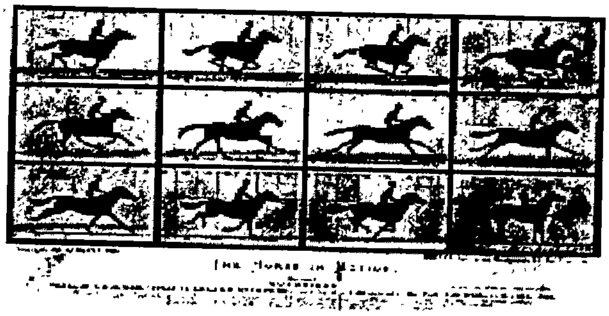


Figure 2.9 The Horse in Motion 1878 Muybridge's film

Eadweard Muybridge, an English photographer, merged a series of 24 still images of a horse into a motion picture of the animal galloping when shown in sequence. Muybridge's film "The Horse in Motion" was commissioned in 1878 as an experiment to test if the horse's four feet could ever be raised off the ground at the same time. Some claim it to be the first silent film ever made. "The first motion image to show sequential action is a 2.11-second short video. Louis Le Prince. a French inventor, directed "Roundhay Garden Scene" in 1888 also illustrated in figure 2.10. Le Prince captured an afternoon in his father-in-garden. Le Prince's brother is shown playing the accordion. The Guinness Book of Records says that this is the world's oldest surviving film. Because technology was limited until the early 1900s, most of the films that followed Le Prince's were short. It wasn't until 1927 that sound was added to motion pictures. The Sony D1 format was the first commercially available digital video format in 1986. H.264 and MPEG-4 are two prominent compressed digital video formats today, in addition to uncompressed ones.

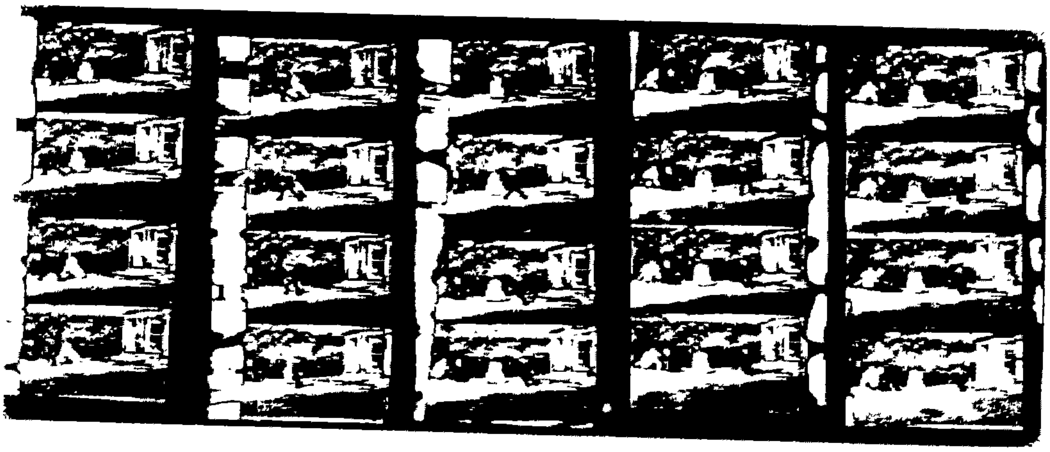


Figure 2.10 The National Science Museum. London. recreated the Roundhay Garden Scene in 1930

Digital videos can be easily duplicated. A digital smart TV or a desktop computer display may stream digital video content saved on a Blu-ray disc or a computer hard drive. A large number of digital audio soundtracks are now available for television programs and movies. Multimedia applications and services are the most extensive network traffic in the modern era. They have established themselves as the primary mode of communication especially for present and future networks in these times of pandemic. Even when network bandwidth and supported bit rates rise, raw video data transmission requires substantial capacity. For efficient audiovisual transmission, video sequences are compressed.

In order to transmit data over limited bandwidth, video coding compresses and decompresses a digital video sequence. By utilizing coding and compression techniques, available bandwidth can be utilized and managed more efficiently. Video compression techniques utilize the fact that video signals are made up of extremely similar spatial, temporal, and frequency sequences. It is feasible to obtain great compression by reducing duplication in these three domains.

Video compression can be lossy or lossless. The lossy compression preserves the video quality but does not achieve high compression ratios, whereas the lossless methods compress the data volume of the initial raw video signal more efficiently at the expense of the perceived quality of the video service. Lossy compression is used for natural images and video surveillance because it allows for the deletion of unnecessary characteristics to conserve storage and transmission resources. The objective of lossy compression is to minimize the coding rate while maintaining visual quality.

The subject of image and video compression has exploded in recent years, with several coding approaches being developed and polished. International compression standards have spurred the development of applications for image and video coding. JPEG and JPEG 2000 are two well-known formats for compressing still images. Since 1984, the International Telecommunication Union (ITU-T) and the International Standards Organization (ISO) have partnered on standardization of image/video coding as illustrated in figure. 2.11.

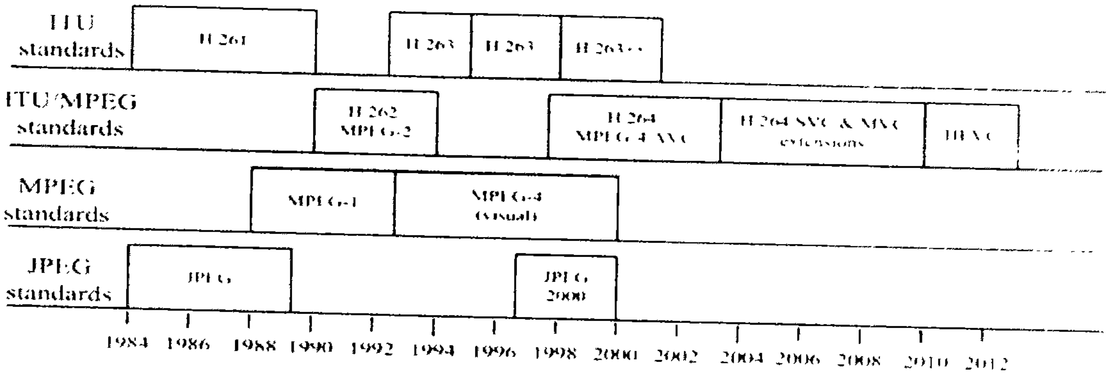


Figure 2.11 ITU-T and ISO/IEC time line of image and video coding standards

2.7.1 Image and Video Compression Standards

Digital images and videos have stormed every mode of communication. They are used everywhere over the Internet, digital photography, medical imaging, remote sensing, surveillance, and facsimile, social media, online advertisements etc. However, the sheer amount of raw data makes storage and transmission challenging. Fortunately, advances in image compression technology have solved the issue. In the mid-1980s, ITU-T and ISO collaborated to create the Joint Photographic Experts Group, a worldwide compression standard for continuous-tone still pictures in grayscale and color (JPEG). The JPEG group resolved to establish a new image compression standard, JPEG 2000, with more flexibility and interchangeability than JPEG. Figure 2.12 illustrates the process of image compression encoding, in which transform is used to decorrelate the signal and quantization is used to reduce the amount of data that must be stored or conveyed. The next sections will provide an overview of the two standards.

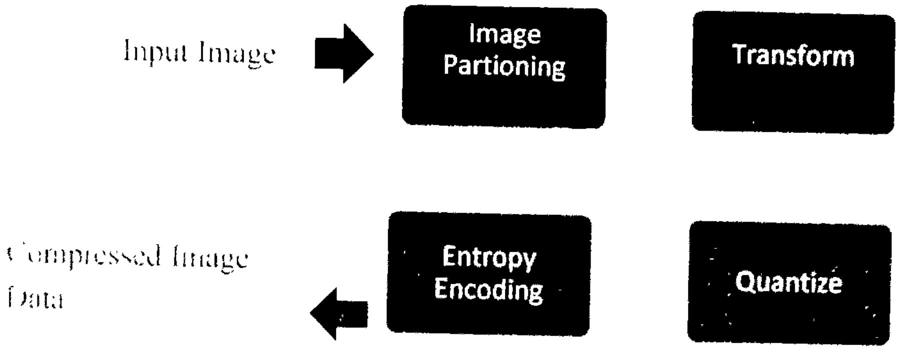


Figure 2.12 General structure of image coding standards

The world today cannot imagine a life without digital videos. Digital videos use the most advanced computer and transmission technology, and compression standards today. Video coding standardization are continuously improving the field of video compression technology. VCEG the Video Coding Experts Group and MPEG the Moving Picture Experts Group the two video coding standards organization merged to create H.262/MPEG-2 in 1994. They formed the Joint Video Team (JVT) in 2001 and developed AVC the Advanced Video Coding Standard also known as H.264 or MPEG-4. They later developed HVEC the High Efficiency Video Coding also called H.265 in 2010.

Further the concept of Inter frame and Intra frame was introduced in to video compression. Intra-frame means that only the information contained within a single frame determines how much data should be compressed. Whereas Inter-frame coding creates key frames that consider the whole image.

1 JPEG

In the mid-1980s, ITU-T and ISO members collaborated to develop a standard for compressing grayscale and color still pictures. This endeavor was dubbed JPEG, with the prefix "joint" denoting collaboration between ITU-T and ISO. JPEG was proposed for effective image interchange across application boundaries. It is used in a variety of areas for storage and transmission, including the Internet, professional and consumer digital photography, and video.

2 JPEG 2000:

Since the introduction of JPEG in the 1980s, a new standard, JPEG 2000, has been developed to address the shortcomings of JPEG. JPEG 2000 does not employ an 8x8 block-based transform, it avoids blocking artefacts. JPEG 2000 is a significant functional improvement to JPEG, offering enhanced low-bit-rate compression efficiency, allowing for bigger image sizes, and simplifying the breakdown architecture. Despite the new standard's greater compression efficiency, genuinely meaningful gain, especially at medium and high levels of quality is not much achieved.

3 H.264/MPEG-4 AVC

In 2003, H.264/MPEG-4 or AVC (Advanced Video Coding) was introduced by cooperative group known as the Joint Video Team JVT and the ISO/IEC MPEG. It has become one of the most frequently used coding standards for high-definition video recording, compression, and distribution.

It uses variable block-size ranging from 16x16 to 4x4, enabling precise segmentation of moving regions; quarter-pixel precision for motion compensation enabling precise description of moving areas' displacements. It uses an in-loop deblocking filter that helps prevent blocking artefacts H.264 video at half the bit rate or less performs much better than MPEG-2 video. It has become the industry standard and is used in a range of video coding applications, such as the iPod and PlayStation Portable, as well as television broadcasting standards. Additionally, H.264 is a well-known codec standard for Blu-ray Discs.

4 HEVC

High Efficiency Video Coding (HEVC) was released as a successor for H.264/MPEG-4 AVC in early 2013. HEVC was intended to provide much greater compression while maintaining a better level of video quality while transmission. HEVC offers twice the data compression ratio and video quality of H.264/MPEG-4 AVC, allowing 8K UHD and resolutions up to 8192x4320. The standard is constantly improving.

2.8 DNA Deoxyribonucleic Acid

Deoxyribonucleic acid (DNA) was found in 1869 by Friedrich Miescher, who was a physician and a biologist. He was born in 1844, in Basel, Switzerland. Miescher was working in his lab on isolating white blood cell protein components. A local surgical facility gave him pus-coated patient bandages. He cleaned them to remove leukocytes and identified proteins inside white blood cells. He discovered a cellular material with chemical features unlike any protein, such as high phosphorus concentration and resistance to protein maceration. He realized, that he had discovered a new substance and claimed that, "a whole family of such slightly varying phosphorous-containing substances will appear, as a group of nucleons, equivalent to proteins".

He tried multiple times to precipitate the solution in different chemicals. This unidentified precipitate dissolved in alkaline solutions but precipitated when neutralized. The material did not dissolve. He concluded it was not a protein. Even after burning this unidentified element he could locate all the usual organic elements namely the nitrogen, hydrogen, carbon and oxygen but not Sulphur an element that is most common in proteins. But it had an appreciable amount of phosphorus that wasn't present in any other biomolecule. Miescher was determined to identify this element, therefore he didn't stop here and further experimented with enzymes that deconstruct proteins into amino acids. Even the proteases had no effect on the precipitate, confirming that it was not a protein. He knew the material was present in the nucleus, therefore he termed it nucleon. The research highlighting his discovery came in 1871.

After gaining professorship, he was overwhelmed by additional duties and had to shift his research emphasis. Other scientists took up the study of DNA. Nucleon was discovered by Nobel laureate Albrecht Kossel, a Hoppe-Seyler lab researcher. Eduard Zacharias, a botanist, linked chromosomes with nucleon, proving that nucleon was a chromosomal component.

Even Nevertheless, the scientific world paid little attention to DNA at the time. This led to a huge disregard of nucleic acids. Among those working on the structure of DNA were Wilkins, Rosalind Franklin and her student Raymond Gosling, and Linus Pauling. Through a combination of luck, ingenuity and inspiration, Watson and Crick reached the finish line first.

After their discovery, three decades of groundbreaking effort followed. In the preface to Friedrich Miescher's collected works, his uncle Wilhelm stated at his death that, "The admiration of Miescher and his work will not wane; on the contrary, it will expand, and his discoveries and concepts will be seeds for a bright future." We are still searching for the mysteries of evolution, health, and the origins of life on Earth, thus Miescher's statements ring true today, and he will always be remembered in the DNA history!

DNA structure representing the different parts of the DNA is explained in the figure below. DNA is composed of a sugar-phosphate backbone and the nucleotide. Each nucleotide is composed of a nitrogen base, a sugar molecule, and a phosphate molecule. DNA may include four different nitrogen bases: adenine (A), thymine (T), guanine (G), and cytosine (C). Figure 2.13 illustrates the structure of a DNA below.

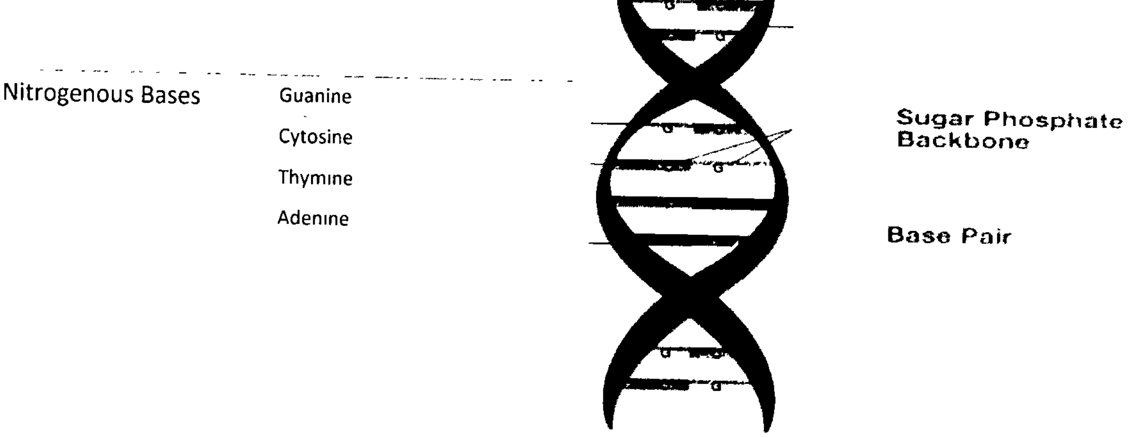


Figure 2.13 DNA Structure [63]

a. DNA Structure

The DNA structure is like a ladder that is twisted like a wire and is called a double-helix demonstrated in the figure below. The DNA molecule is made up of nucleotides, which are made up of sugar, phosphate groups, and nitrogen bases. Sugar and phosphate groups join each DNA strand. It contains adenine, thymine, guanine, and cytosine. A pairs with T, and C with G. The genetic code is determined by nitrogenous bases. Figure 2.14 illustrates the components with in a DNA structure, each is composed of a phosphate, a nitrogen base and deoxyribose sugar.

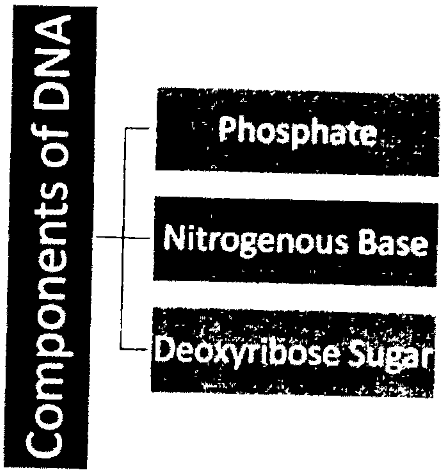


Figure 2.14 Components of DNA Structure [63]

Adenine (A), thymine (T), cytosine (C) and Guanine (G) are the four nitrogen bases that make up DNA. Purines A and G, pyrimidines C and T. The DNA strands flow counterclockwise. The hydrogen connection between the two complimentary bases holds these strands together. Each strand creates a right-handed coil, with 10 nucleotides per turn. Each helix has 3.4 nm pitch. Thus, the distance between two successive base pairs (opposing strand hydrogen-bonded bases) is 0.34 nm. The DNA coils up into chromosomes, each containing one molecule of DNA. Humans have around twenty-three chromosomal pairs in their nucleus. [60]

b. DNA Computing

DNA computing makes use of biological molecules, atoms, or enzymes to do calculations on a computer or as a storage medium in place of conventional silicon chips. This novel idea of using individual molecules or even atoms for computing, stems all the way back to 1959, when American physicist Richard Feynman unveiled his nanotechnology concepts. However, physical realization of DNA computing occurred in 1994, when American computer scientist Leonard Adleman demonstrated in his paper Molecular Computation of Solutions to Combinatorial Problems the experimental use of DNA in solving computational problems. He used a seven-point Hamiltonian route problem, often known as the travelling salesman problem, to test his experimental hypothesis. In this task, the salesperson must determine the quickest route between seven cities whose distances are known, while avoiding crossing any city twice and returning to the originating location. Adleman depicted each city with a short DNA sequence of around 20 bases and a corresponding strand of the same length as the roadway connecting them. Not all of them were right, and he spent the next week extrapolating and filtering out the shortest route using a variety of strategies but his experiment opened doors to a diverse field of DNA use in computing.

DNA-based computing is a kind of computing that utilizes DNA molecules rather than digital logic circuits. The biological cell is seen as a sophisticated computer like entity. The four amino acid bases that make up DNA, typically denoted by the letters A, T, C, and G, are employed as operators in the same way that computers utilize the binary numbers 0 and 1. DNA molecules are encoded according to the instructions of the researcher and then induced to recombine, resulting in billions of simultaneous "calculations." The field is still in its infancy, and its potential is being explored.

c. DNA Steganography

In the year 1999, Clelland et al. [62] introduced a technique to hide a secret messages using DNA. They turned a message into a string of quaternary digits and then replaced it with a sequence of nucleotides, they introduced double steganography by further hiding the encrypted DNA in to microdots. Figure 2.15 illustrates a sample of the Cleland's encryption technique.

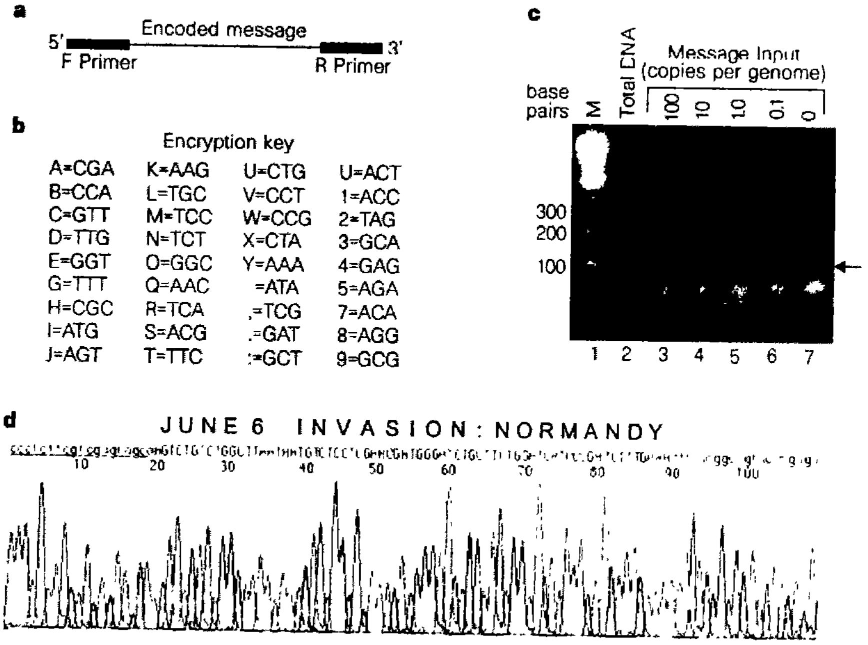


Figure 2.15 Cleland's encryption technique [62]

DNA is also being used in the field of watermarking. Conventional DNA watermarking techniques encode data in the form of DNA codes that are subsequently placed into the genome as DNA barcodes or embedded in DNA fragments to conceal the data.

When DNA is used for data embedding, the basic approach involves starts with the use of a real DNA as a key. For this researchers choose any DNA of any suitable size appropriate for their scheme. Now there are many approaches to using this DNA code. One approach is to use this DNA code as a key for creating either a fake DNA by embedding their secret message in it. The secret message may be a plain text, audio message or image of any length. The size of the secret message also varies depending upon the purpose for it is used. Now this embedding in DNA is not done just as it i.e. anywhere or any place. There are algorithms that can be used for embedding. Devising such algorithms is an active area of research. In order to add further security this fake DNA is not transmitted as it. It is embedded in a cover medium. Cover medium could be an image, watermark or a video. There are thousands of algorithms devised by researchers for embedding the fake DNA in to the cover medium of the chosen type. The embedding of DNA in a cover medium is an active area of research these days. The purpose of these algorithms is to hide the DNA in the cover medium in the best possible way so that it cannot be detected by a malicious attacker who would detect the hidden message and use it for their purpose. Once the message is embedded it can be transmitted through any mode of transmission of information i.e. either over the internet or through any physical device. The second approach is to use the DNA in devising an algorithm for embedding in the cover medium. This is again another an active area of research where thousands of researchers gather up every day for devising not just a secure but also an efficient embedding technique in terms of time, space, complexity and many other criteria depending upon the purpose for which this process is carried out.

DNA computing is an active area of research with a lot of scope at every step of processing. That is why this field of study is gaining increasing interest from both biologists and computer scientists as they have discovered that there are several prospects for extending and changing DNA properties and functions in order to address real-world challenges.

2.9. Literature Review Video Steganography

DNA characteristics are suggested by Kar et al. [4] for embedding secret message in a video. Selection of frames is based on the binary values of DNA, the number of codons, and a user-supplied key. For frame selection, a random number is generated by the linear congruential generator and then XORed with DNA. Burger chaotic map in combination with LGC i.e. linear congruential generator is utilized for pixel selection. LSB replacement strategy is used for concealing the data at arbitrary positions. In this scheme the selected frame is ultimately a random frame that might not be detected. But hiding data in random frames does not guarantee safety of the hidden message. Hiding in any random frame lacking analysis of the contents of the frame host artifacts in the frame that make the video itself perceptible.

The modified least significant bit strategy of Ramalingam et al. [5] creates an encrypted and a non-encrypted avi file. The encrypted file contains the encrypted message using a symmetric key for encryption of text. Message is revealed by finding out the bit by bit difference between the two created files and then decrypting the message using a symmetric key which is altogether a compromised approach in terms of encryption using a symmetric key and using two files for transmission.

Cao et al. [6] suggested an embedding method based on the H.264 that uses motion vectors to store the secret message. To reduce the changes in motion vectors, linear block codes have been used. But this technique compromises the visual quality of the reconstructed images.

Bin et al. [7] introduced a steganography approach based on a motion vector through the use of matrix encoding. A motion vector component with a large amplitude that exists in both vertical and horizontal components are chosen to embed the coded message. The Human Visual System is capable of detecting changes in a slow moving object, but failing while when the same object travels quickly. Hence author utilizes large-scale motion vectors for the purpose of inserting the hidden message. No encryption of the embedded message has been proposed in this method.

Kelash et al. [8] concealed data within frames by splitting each pixel of the frame into two parts. The data is embedded into the pixels based on their color histograms. The frames chosen are on the bases of high threshold difference between RGB histograms of two frames. The pixels are embedded in a frame with a greater difference. This gives a clear clue to an eve's dropper when he sees these mild color changes due to pixel embedding compared to sharp color changes in a similar video.

The DNA insertion algorithm has been modified by Malathi et al. [9]. The algorithm makes use of two distinct keys. The message is encrypted using the first key. K2 being the second key is produced randomly and it segments the DNA into equal parts. At the beginning of each segment, the resulting cypher characters are inserted as binary bits one by one. A DNA rule-based dictionary is used to translate the binary sequence into DNA bases.

Wang et al. [10] encrypt the secret message prior to embedding using the vigenere cypher. The cypher text is then converted in to binary and hidden by means of DNA encryption approach, resulting in a better-quality video steganography technique and detection of modified frames in a stego-video. Jiao et al. [11] proposed using encryption to conceal data within living beings' DNA. The message is binary transformed from DNA bases or codons. 35 codons were employed in the implementation of this study. The position of the codons that are substituted by the message is presumed at first. The encryption allows for quiet DNA sequence mutation. The complement of encryption is performed during the decryption process in order to recover the hidden message.

In a cloud computing context, Mohammad et al. [12] suggested a data concealing approach in DNA sequences for resource sharing. This document protects data in the cloud that is accessible to the public. This approach uses complementary algorithm to combine the bases of used DNA and that of the DNA transformed message to form a complete message. The fictitious DNA is manufactured and stored in the cloud. The client downloads the false DNA and retrieves the data using the inverse of the complimentary process. The complimentary algorithm has a higher likelihood of cracking than the substitution approach.

According to Peterson et al. [13], the confidential message is concealed within a DNA sequence by substituting the character on the free consecutive bases of a DNA sequence. They assigned a codon to every letter. In case a letter appears multiple times it will be easy to decrypt. This characteristic can be used by an attacker to decrypt the communication. This is one of the pioneering works of concealing data within a DNA sequence and uses a very preliminary approach.

Shiu et al. [14] concealed a secret message within DNA using insertion, substitution, and complementary techniques. Using the insertion approach, the message i.e. transformed in to a binary message is placed at the beginning of the binary transformed DNA sequence. This work likewise use the replacement approach in conjunction with the complementary rule to embed the concealed message in the sequence. The cracking probability of each approach is determined to ensure the algorithms' security. The insertion procedure is the least likely to result in cracking.

Agarwal et al. [15] employ substitution method using a DNA dictionary to camouflage the data inside the DNA. The proposed strategy improves the embedding capability and security of the present substitute method. Codons are utilized to conceal information inside the DNA sequence. Binary transformations are performed on the codons, and their lengths are tested for multiples of six. If it is a multiple of six, it is left alone; otherwise, it is appended to the binary string with additional zeros. Codons are used to convert the six-bit binary code to DNA. The altered DNA is then transmitted to the recipient. The receiver retrieves the data by reversing the algorithm.

Mumthas et al. [16] use 2D DCT steganography, Rivest–Shamir–Adleman (RSA), random DNA encryption, and Huffman encoding. The original message is converted to cypher text using the RSA technique, then random DNA encryption and compression are applied. A random DNA encryption technique is utilized instead of fixed DNA encryption. The mapping of codons against amino acids is through DNA-based random numbers. The positions of codons and amino acids are decided by a random permutation of accessible places. Each 8x8 block of the cover video's frame is subjected to DCT. The compressed message is concealed in the coefficients' less significant sections.

Khalifa et al. [17] suggested a steganography algorithm using DNA also. The proposed approach is divided into two distinct stages. In stage one, the secret message is originally encrypted using a DNA-based Playfair cypher. At stage two, the encrypted secret message is concealed within some reference DNA via a substitution approach. That is, the reference DNA's bases are substituted with the encrypted DNA in accordance with a generic two-by-two complementary rule. The performance of the presented algorithm is evaluated in terms of its ability to conceal data as well as its resistance to attacks. Hiding data in DNA makes advantage of a variety of biological properties and is not particularly efficient to execute economically.

To encrypt and disguise the user data in the video, Jose et al. [18] proposed a combination of cryptography and steganography. For generating keys, they developed SHA512-ECC, which combines the SHA512 hash function with the elliptic curve -ECC. They employed an upgraded version of the crow search algorithm, which they called CM-CSA, for pixel selection. The plain text is first taken as input, after which it is compressed and encoded using an improved version of the Huffmann Algorithm known as the IHA. SHA512-ECC algorithm is used to encrypt the DNA encoded data. Finally, these encrypted data is hidden into the optimal pixel points of the video frames using the CMCSA.

Faud et al. [19] embedded text in the object motion regions by using motion analysis. Using DCT-psycho visual effects of hiding messages, the suggested technique picks six DCT coefficients in the intermediate frequency. The proposed approach is used to embed a message by altering middle DCT coefficients. Text is not encrypted which recovers the message once pixels are identified.

algorithms, embedding and extracting algorithms. Compression of secret data is done using ZIP algorithm. Encryption is done using AES algorithm. The compressed and encrypted secret data is embedded at the end of the frame evenly to produce stego FLV. This technique is only tested for flv codecs.

Husein et al. [21] proposed an encryption using Turbo Code. Embedding is done using least two Significant Bit Technique. This strategy embeds a logo inside video frames by using the Turbo system and least two significant bit. It embeds the logo after converting the cover to the frequency domain by using Fast Fourier Transform. Although the technique promises good results but degrades the video quality.

Karmakar et al. [22] use a dictionary to describe each video frame as a sparse matrix with only few substantial sparse coefficients. They compressed the video by encoding non-zero sparse coefficients along with their addresses. A 5D hyper-chaotic system is used in conjunction with DNA coding to generate a hyper-chaotic key, which is then incorporated into the encryption scheme to increase the security of video transmission. Due to the use of patch-based modification, the rebuilt frames exhibit a blocky appearance.

In the method used by Rout et al. [23], motion vectors were used to represent the displacement of the spatial video unit from one frame to reduce the temporal redundancy. To reduce inter frame temporal redundancy, macro blocks (MB) of B×B pixels were used in the motion model in the video compression methodology. The difference (error) frame in motion compensated frame prediction was determined as the difference between the actual frame and the predicted frame. There are often smooth regions with less motion in original video frames. As a result, the motion vectors and corresponding prediction error blocks are mostly homogeneous in these regions. These regions were targeted for embedding in this paper.

The aim of method proposed by Z.S Younus et al. [24] is to hide data in video and to ensure its security. This method begins with writing the message using the alphabets of English language then secured by encoding. During embedding, knight tour algorithm and LSB method was used. This process breaks the video into frames, these frames are then altered into images. Next step involves random selection of frames to be used as cover, knight tour algorithm was used to select pixels from selected frames. LSB method was used for hiding the cypher text in the selected pixels. After embedding, extraction process starts. The recipient divides the stego video into frames. Then using the same steps and knight tour algorithm used for embedding, pixels containing the data are identified and secret message is extracted using LSB method.

Method of I. P. Febin et al. [25] focuses on identifying violence or fights by filtering movements and motion boundary SIFT (MoBSIFT). Motion boundary SIFT (MoBSIFT) was an improved method for detecting violence. Extracted frames were converted to gray scale. Motion in each frame was detected through movement filtering. Only frames having motion will undergo feature extraction. Temporal derivative was an efficient method used for identifying violent videos. This research paper also discusses how nonviolence videos can be easily identified. There was still a need for more accurate method for feature selection.

Suresh et al. [26] devised a new method to hide data by the application of Fractional Grey Wolf Optimization. The optimization algorithm reads the fitness function derived from the cost function to find the global optimal region to embed the secret information. Encryption is accomplished through two steps: key generation and nonlinear diffusion.

Wan et al. [27] proposed method for removing redundant frames for video surveillance. His approach begins with the determination of the spatiotemporal interest points of each frame using the improved spatiotemporal interest point detection algorithm. Surround inhibition was then used in conjunction with local and temporal constraints to identify static interest points in the frame. In this way, by comparing the changes in the number of motion detection boxes, a large number of redundant frames in a long video can be removed which increases the speed of processing. Video segmentation was performed on the long video which doesn't have redundant frames and then SOI was extracted for retrieval of video event.

In Huang et al. [28] method, CapsNet was used to extract spatiotemporal features and generate inter-frame motion curves from video sequences. The purpose of this approach was to generate a summarized video for which key frames were selected. The key frames were chosen using the automatic shot segmentation method and the self-attention model.

Zhang et al. [29] proposed a technique for selecting pixels for embedding using a particle swarm optimization-based bacterial foraging algorithm. Zhang et al. [28] also suggested using Hurst Component for selecting appropriate pixels for embedding audio watermark.

Raju et al. [30] use the Mersenne Twister algorithm to hide information inside random pixels. They embed the AES encrypted test into the LSB locations achieving a PSNR 71.482 with MSE of 0.0046. The data is embedded in random pixel locations without looking at the contents of the frame, which generates visible artifacts in the stego video. A concise review of these techniques is given in table 2.2 below.

Table 2.2 Literature Review Videos Steganography

Paper Title	**Approach**	**Analysis**
Nirmalya et al. (2018)[4]	Proposed a technique using DNA for hiding data in a video frames. Frame selection is done using arbitrary DNAs and a user given key.	This technique suggests random frame for embedding without looking in to its contents that can produce artifacts in the video generated due to insertion of encrypted data
Eltahir et al. (2009) [5]	Suggests LSB method with an augmented secret message size approach.	RGB(3, 3, 2) is used

Ramalingam et al. (2011) [6]	Uses modified least significant bit approach and symmetric key	It is altogether a compromised approach in terms of encryption using a symmetric key and using two files for transmission
Cao, Zhang et al. (2015) [7]	Recommend embedding the secret message in motion vectors.	This technique compromises the visual quality of the reconstructed images Once the pixels are detected the plain text message can be easily retrieved
Bin et al. [8]	Proposed matrix encoding of motion vectors.	Once the pixels are detected the plain text message can be easily retrieved
Kelash et al. (2013) [9]	Data is embedded directly in frames. Pixels are embedded w.r.t their color histograms after division into two portions.	Produces perceivable artifacts in stenographic video
Malathi Pa et al. (2017) [10]	Propose the modification to the DNA insertion procedure, text is encrypted using fake DNA and two keys.	This technique reduces video imperceptibility.
Jiao et al. (2009) [11]	Using mutations of codons data is hidden inside the DNA and a fake DNA is generated and uploaded on cloud.	A symmetric key is a compromised approach.
Peterson et al. (2001)[13]	Suggested embedding in DNA to create a fake DNA. They used a total of 64 symbols for encodings.	An attacker's major objective in cracking the encoded message will be to exploit the property of often occurring letters
Shiu et al. (2010)[14]	Proposes a reversible data hiding scheme for hiding a secret message inside DNA.	The resultant image quality is. Significantly lower.

Rishi Agarwal et al. (2014) [15]	They proposed DNA based substitution using dictionary.	The algorithm increases the embedding capacity and security.
Mumthas et al. (2017) [16]	Used 2D DCT steganography and an encrypted text.	The initial message is turned to cypher text using the RSA algorithm and then compressed using random DNA encryption. Random DNA encryption is used.
Amal Khalifa et al.(2012) [17]	Uses DNA substitution using a 2 by 2 complementary rule	
Jose Subramaniam (2020) [18]	Proposed a combination of elliptic curve and SHA-512	For pixel selection, they used an improved version of crow search algorithm.
Faud Ernawan (2020) [19]	Used motion analysis to embed text in the object motion regions	They embedded the message in the middle DCT coefficients of pixels.
Younas Younas (2020) [20]	Encrypted text using random numbers and mathematical equations. Embedding is done using the knight tour algorithm and LSB method	The serial selection of pixels within the frame that is used for embedding the information inside it. This makes the method vulnerable to electronic attacks that can detect the hidden data by analyzing the pixel patterns
Arraziqi Haq (2019)[21]	compressed text using ZIP algorithm before encryption through AES	Embedding is done evenly in frames
Husein Jum Al-Thab (2020) [22]	proposed an encryption using Turbo Code Embedding is done using Least two Significant Bit Technique	The high computational complexity of the turbo encoder and decoder, which may affect the performance and efficiency of the method. The low embedding capacity of the L2SB technique, which limits the amount of secret data that can be hidden inside the video frames.

		The sensitivity of the L2SB technique to various attacks, such as compression, noise addition, filtering, etc., which may degrade the quality and security of the hidden data.
Karmakar et al. (2021) [23]	A five-dimensional hyper-chaotic system was employed in conjunction with DNA coding.	The 5D hyper-chaotic system and DNA coding were used to develop a hyper chaotic key, which was then put into the encryption scheme to boost the security of video transmission.
Rout et al. (2020) [24]	Elliptic Curve Cryptography was used to encrypt stego keys, and a threshold method was employed to decide the quantity of secret data to encode in randomly picked frames.	Utilizes the multiscale directional transform to incorporate secret data; frames from the cover video are transformed using the FDCT. ECIES is used to encrypt the stego key.

2.10 Literature Review Frame Selection in Videos

The aim of method proposed by Z.S Younus et al. [25] is to hide data in video and to ensure its security. This method begins with writing the message using the alphabets of English language then secured by encoding. During embedding, knight tour algorithm and LSB method was used. This process breaks the video into frames, these frames are then altered into images. Next step involves random selection of frames to be used as cover, knight tour algorithm was used to select pixels from selected frames. LSB method was used for hiding the cypher text in the selected pixels. Meanwhile, selected images also undergo the same process which re then converted into frames and then a stego video was formed. After embedding, extraction process starts. The recipient divides the stego video into frames. Then using the same steps and knight tour algorithm used for embedding, pixels containing the data are identified and secret message was extracted using LSB method. Method of I. P. Febin et al. [26] Theresa Joy focuses on identifying violence or fights by filtering movements and motion boundary SIFT (MoBSIFT). Motion boundary SIFT (MoBSIFT) was an improved method for detecting violence. Extracted frames were converted to gray scale. Motion in each frame was detected through movement filtering. Only frames having motion will undergo feature extraction. Temporal derivative was an efficient method used for identifying violent videos. This research paper also discusses how nonviolence videos can be easily identified. There was still a need for more accurate method for feature selection.

Meenu Suresh and I. Shatheesh Sam [27] devised a new method to hide data by the application of Fractional Grey Wolf Optimization. Most important step was the extraction of key frames or frames which contain secret or very important information. Frames having more similarity are selected as key frames. Optimal frames were selected using optimization algorithm that enables effective concealment of the secret information. The optimization algorithm read the fitness function that was derived from the cost function to find the global optimal region to embed the secret information. Encryption was accomplished through two steps: key generation and nonlinear diffusion. On the cover image, the wavelet transform was used for embedding then Inverse lifting wavelet transform (ILWT) was applied. The secret data was extracted from the optimal region using an optimization algorithm.

Accordong to Shaohua et al. [28] long videos contain many redundant and meaningless frames that must be detected and removed. The method proposed begins with the determination of the spatiotemporal interest points of each frame using the improved spatiotemporal interest point detection algorithm. Surround inhibition was then used in conjunction with local and temporal constraints to identify static interest points in the frame. In this way, by comparing the changes in the number of motion detection boxes, a large number of redundant frames in a long video can be removed which increases the speed of processing. Video segmentation was performed on the long video which doesn't have redundant frames and then SOI was extracted for retrieval of video event. This section combined low-level features like contrast, sharpness, and color with advanced semantic features like attention and face information. Each frame of the video was converted to a grayscale image, which was then processed using low-pass filtering. According to empirical testing, Attention, Contrast, Colorfulness, and Sharpness are critical feature elements for video segmentation. Facial information is critical; however, not every human face appears in every video; thus, an influence factor η was added to the Facial score. The lengthy video was divided into segments. These video segments contained the key frames from the original video that were used to generate video subtitles. Local features were extracted by extracting the features of the frame at a specific time, global features were extracted by extracting the SOI features, and temporal endpoint features were extracted by extracting the SOI moment features. A deep convolution network was used to extract the high-level features of each frame, and then average pooling was performed on the video features within the SOI, that is, averaging all frames in the SOI, to construct local and global video features. The scene events in the video were confirmed by matching them. This article employs the VGG model, which has been pre-trained on ImageNet, to extract local, global, and temporal endpoint features from frames. For transforming question texts, the question was divided into words using punctuations and spaces, then these words were transformed into a 300-dimensional word vector with word2vec. In the end, to extract language features the word vectors were sent to the LSTM.Then visual features and test vectors were combined. In that paper, the target combination feature vector and the target relationship generated by the image appearance relation model were used to provide image information for the image.

In the method used by Shuvendu et al. [29], motion vectors were used to represent the displacement of the spatial video unit (macro block) from one frame to the next so that temporal redundancy can be reduced. To reduce inter frame temporal redundancy, macro blocks (MB) of B×B pixels were used in the motion model in the video compression methodology. The difference (error) frame in motion compensated frame prediction was determined as the difference between the actual frame and the predicted frame. There are often smooth regions with less motion in natural video frames in natural video frames. As a result, the motion vectors and corresponding prediction error blocks are mostly homogeneous in these regions. These regions were targeted for embedding in this paper to mask the embedding distortion caused by changing the motion vectors. One of the primary goals of the proposed MV-based embedding scheme was to reduce embedding error to the point where prediction error-based steganalysis attacks are less effective against the proposed scheme. During embedding, standard motion estimation method was applied for selecting the macro blocks (MB) which fulfil the requirement. Homogeneous regions around the selected MBs were identified. Homogeneous region was used to select alteration locations (say MBs) .Then these MBs were divided in two different groups to ensure efficiency in embedding. Candidates for embedding are high-value motion vectors from homogeneous image regions. To achieve this goal, motion vectors with absolute values greater than 7 are chosen for embedding. Depending on the polar orientation of the motion vector, macroblocks are divided into two groups as odd and even valued group. During motion-vector embedding, by modifying the motion vectors, a secret bit stream (W) was inserted with the video sequences. After the bit is altered, quantization and bit compression were performed to generate the encoded video bit-stream. Then extraction process, which is reverse of embedding was carried out. In distortion analysis, embedding was accomplished by modifying the motion vectors during video compression. To replace the originals, the motion vectors of similar regions were used. As a result, the new PE will be identical to the originals. In complexity analysis method, the alteration is carried out with the MVs. For selection of the proper MV in the region of 11 × 11 window, full search algorithm was used.

In Cheng et al. [30] method, CapsNet was used to extract spatiotemporal features and generate inter-frame motion curves from video sequences. The key frames were then chosen using the automatic shot segmentation method and the self-attention model. The length of the input and output vectors characterized the probability that the entity exists, and the orientation of the vector represented instantiation parameters such as position, orientation, size, speed, and color, among others. A nonlinear function known as the squashing function was used to ensure that the length of the short vector can be reduced to almost zero, while the length of the long vector is compressed to close to but not exceeding one. Then the transition effects detection (TED) method was applied. In this method, within local sliding windows, a method based on G1-continuity error and cubic H-Bézier curve fitting was used. Because the transition effects are mostly within 15 frames, sliding windows with a length of 10 and a stride of 5 were built. As a result, 5 frames were calculated each time. A transition can thus always be detected within three sliding windows. In each sliding window, a cubic H-Bézier curve fitting was performed. All information in the sequences could be accessed directly, and information from the source sequences could be passed directly to all steps in the decoding process. To number the position of each vector, a positional embedding method was added. A shot's key-frame sequence was used for both stable optical flow estimation and video motion summarization. And the highest value point on the attention curve was chosen as the shot's sub-summary. Finally, those shot sub-summaries were concatenated to the overall video content summarization. [31] Z Younus et al. proposed a method for data hiding in video using the least significant bit (LSB) method and improving it by using the knight tour algorithm to conceal the data inside the AVI video file and a key function encryption method to encrypt the secret message.

Nirmalya et al. [4] suggest a mechanism based on DNA characteristics for embedding secret info in a video clip. Frame selection is based on the binary values of DNA, the number of codons, and a user-supplied key. For frame selection, a random number is generated by the linear congruential generator and then XORed with DNA. Burger chaotic map is used in conjunction with the linear congruential generator to choose pixels, concealing data at arbitrary points using the least significant bit replacement approach. The modified least significant bit strategy of Ramalingam et al. [5] outperformed Eltahir's approach. Cao, Zhang, and colleagues [5] proposed an embedding method based on the H.264 compression technique, which involves the secret message being stored in motion vectors. Additionally, it makes use of linear block codes to reduce the amount of time that motion vectors are altered. However, this technique degrades the visual quality of the rebuilt images, thereby violating the fundamental rule of concealment by displaying vivid evidence of a steganographic film. Bin et al. [7] suggested a steganography technique based on a motion vector via matrix encoding. To embed the coded message, a motion vector component with a large amplitude that exists in both vertical and horizontal components is chosen. The Human Visual System is capable of identifying changes in a slowly moving item but fails to do so when the same object moves rapidly. As a result, the author employs large-scale motion vectors to insert the secret message. This approach makes no attempt to encrypt the embedded message. Kelash et al. [8] directly inserted data into video frames by separating each pixel into two pieces and embedding data into the pixels based on their color histograms. The frames were chosen based on their high threshold difference between their RGB histograms. The pixels are embedded in a frame that has a larger disparity between them. This provides a clear indication to an eve's dropper when he observes these subtle color variations caused by pixel embedding in comparison to the abrupt color shifts in a comparable film. A concise review of these techniques is given in table 2.3 below.

Table 2 3 Literature Review Frame Selection in Videos

Paper Title	**Approach**	**Analysis**
Meenu Suresh , I. Shatheesh Sam (2020) [27]	Fractional Grey Wolf Optimization	The Structural Similarity Index was used to extract the essential frames in this work (SSIM). Individual key frames were then subjected to region creation utilizing grid lines. We applied the optimization algorithm. Simultaneously, the secret data was encrypted to increase the suggested method's security. (LWT) was employed to obtain the

		wavelet coefficient on the key frames.
Cheng Huang and Hongmei Wang (2020) [30]	A new method for summarizing video content as well as video motion was developed.	Capsules Net was trained to extractor the spatiotemporal features. The extracted features were used to construct an inter-frames motion curve. Following that, a method for automatically segmenting video streams into shots was developed using transition effects. Key-frame were chosen using Self-attention model for video motion summarization.
Shaohua, Xiaolong, Tian and Zonghua (2021) [28]	proposed a superframe segmentation-based long video event retrieval algorithm	By recognizing the motion amplitude, a huge number of unnecessary frames were effectively deleted from the lengthy film. Using a super frame segmentation technique the video is segmented into SOIs that featured the video events.
Shuvendu Rana , Rohit Kamra and Arijit Sur [29]	Proposed embedding in motion vectors	To increase imperceptibility against typical steganalysis approaches, a fast search window and an embedding strategy based on polar orientation were applied. A number of experiments was done to demonstrate the proposed scheme's superiority over other relevant steganographic methods.
I. P. Febin, K. Jayasree and Preetha Theresa Joy (2020)[26]	SIFT (MoBSIFT) and movement filtering was used.	Video were filtered using this way by a temporal derivative-based movement filtering algorithm to avoid feature extraction on the majority of nonviolent behaviours. Only

		the filtered frames were allowed to proceed to the feature extraction stage. To create the MoBSIFT descriptor, the motion boundary histogram was retrieved.
Zeyad Younus and Ghada Younus (2020) [29]	This paper proposed to conceal the data inside the AVI videos.	Knight tour algorithm and LSB was used for embedding.

2.11 Video Forgery Detection Insight

Digital video forgery detection has been utilized to ascertain the validity of digital video forms. These video forgery detection techniques can also be used to identify faked frames in a stego video.

These strategies are classified into active and passive approaches. The active approach is primarily concerned with unseen data and requires information such as a watermark or fingerprint to be pre-embedded into images. It does so by verifying the integrity of the pre-embedded data. In the absence of pre-embedded data, the latter approach is more acceptable in certain instances, such as those involving video, photo images, or audio. Passive techniques are characterized as splicing, source identification, and copy-move forgeries [41], [42]. These techniques are used to detect video tampering in digital video and double compression formats like as MPEG or H.246. However, because the majority of movies lack pre-embedded information such as a watermark or signature, an active method makes it difficult to detect manipulation. As a result, the scientific community has placed a premium on passive video forgery detection approaches in recent years. To understand the techniques and terminologies of video forgery the section below defines the fundamental terms required to comprehend this survey.

2.11.1 Video forgery techniques

Video forgery techniques are categorized into two subclasses i.e. the Intra-frame forgery and Inter-frame forgery. These forgeries can be carried out with the aid of variety of video editing softwares such as Adobe Photoshop. The many types of digital video forgeries are depicted in Figure 2.16.

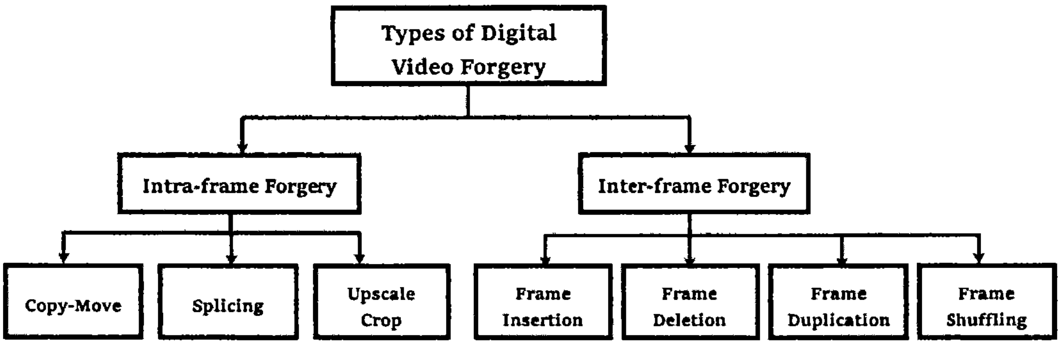


Figure 2.16· Digital video forgery

I. Intra-frame forgery

Also known as the spatial video tampering. This alters the original contents of selected frames. The following examples illustrate types of intra-frame forgeries.

a. Forgery Copy-Move

This sort of forgery inserts or deletes any object or duplicates objects in a video by copying and pasting them to another spot within the same or a different frame of the movie. They can also be used to conceal an area within the frame [37-46]. Figure 2.17 illustrates a (a) a video scene where a flower is copied and pasted into another location within the same video frame. In (b), the video frame the keyboard present in the original frame is removed. There are two methods of frame forgery:

b. Temporal Copy and Paste

It fills in forged spaces in the video frame by using similar pixels from neighboring sections of the frame or the most lucid blocks from adjacent frames. Figure 2.17 below illustrates a video frame region in which a flower is copied to another location from the same video frame. In scene b, an original video of a monitor with accessories is shown and in the adjacent image, the real video frame is devoid of a keyboard. [48].

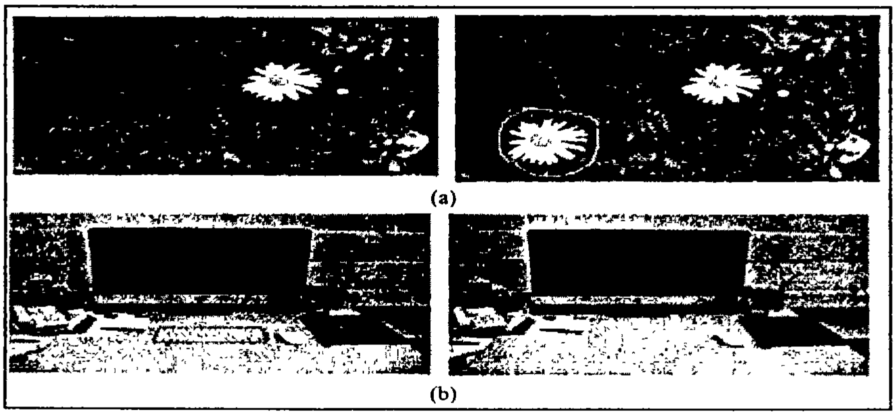


Figure 2.17. video frame region in which a flower copied to another location from the same video frame. In scene b, the real video frame is devoid of a keyboard. [48].

c. ETS Texture Synthesis

In this type of forgery missing areas of a video frame are filled with sample textures.

. Splicing:

A new video frame is manufactured in this sort of forgery by photocopying and pasting a fragment from one video frame onto another. Figure 2.18 illustrates a video splicing forgery in which the objects of two video frames are combined to create a new prepared video frame.

. Upscale Crop:

An upscale crop removes the outside area of the video frame in order to eliminate a region or object [48]. Figure 2.19 a and b show the actual video frame and its upscale crop where a lady is cropped from the video.

II. Inter-frame forgery

Also called temporal manipulation, which changes the arrangement of the frames in a video. Figure 2.20 illustrates inter-frame forgeries. Various types of inter-frame forgeries are discussed below

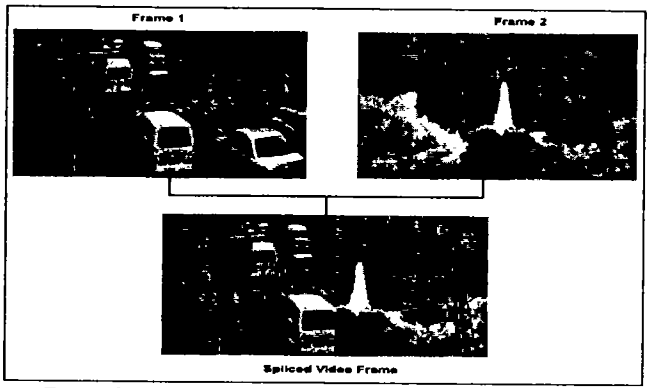


Figure 2.18: Video splicing forgery combining two frames

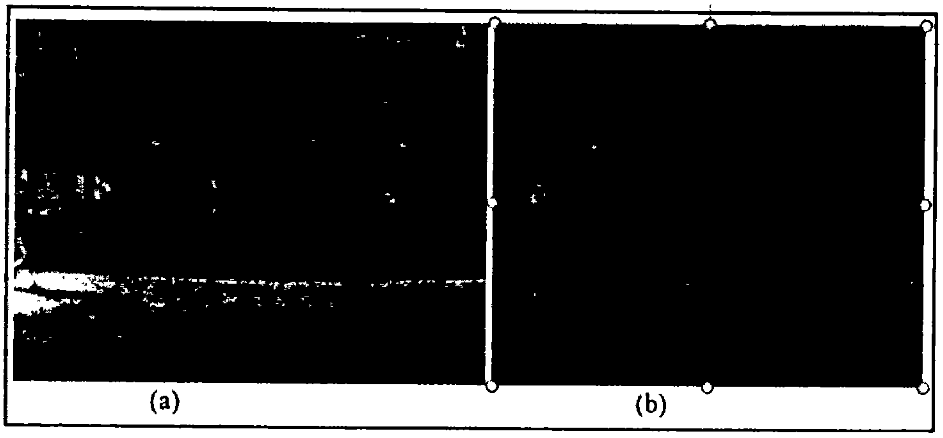


Figure 2.19: (a) the original frame (b) frame shows a lady is cropped from the video. [48]

a. Frame Deletion:

This sort of modification intentionally eliminates frames from a video in order to fabricate evidence of unlawful activities. Figure 2.20 illustrates a video sequence after deleting third and fourth frames from the original video.

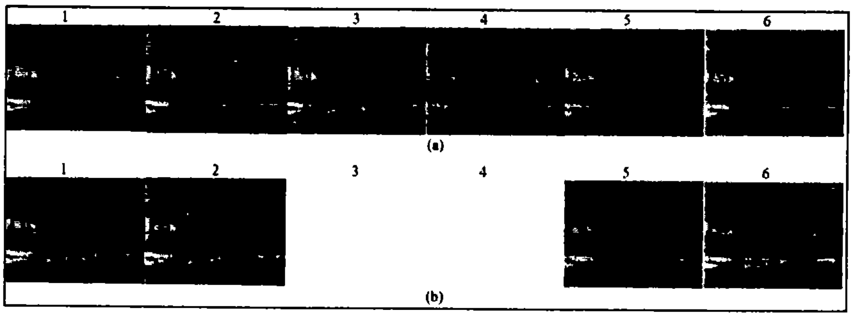


Figure 30· (a) Original video (b) 3rd and 4th frame is deleted from the video [48]

b. Repetition of Frames:

This sort of counterfeit deliberately copies portions of a video. Figure 2.21 shows a video sequence created by duplication frames in the video.

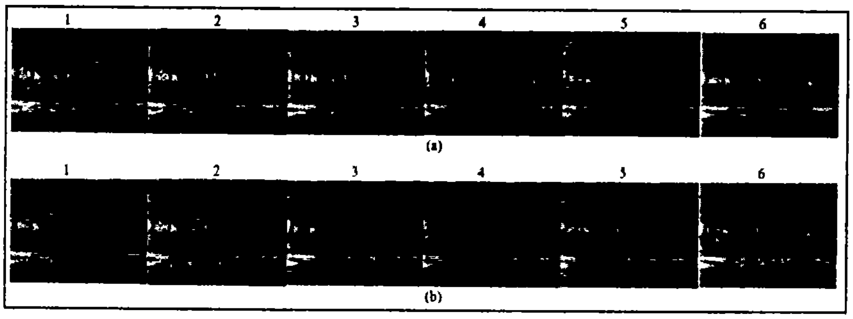


Figure 2.21. (a) Original video sequence (b) Forged video after 6th frame duplication with the 3rd frame [48]

c. Mirroring of the frame:

This forgery replicates some regions of the video and pastes them at random positions inside the same video. In Fig. 2.22 shows the forged video sequence created by copying a mirrored duplicate at M2 i.e. second frame, and a mirrored copy at M6 i.e. sixth frame.

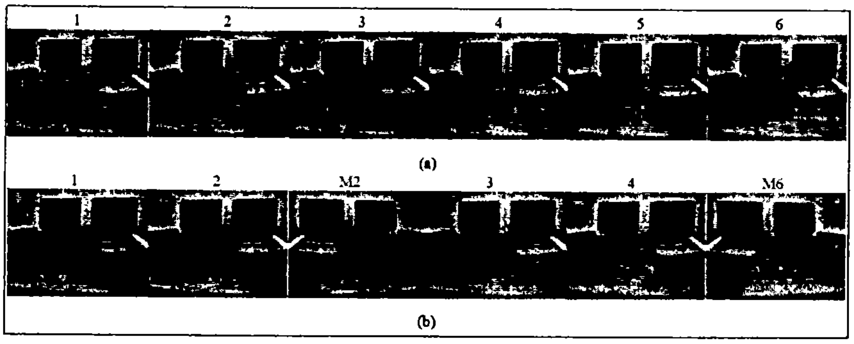


Figure 4.22: (a) Original frames (b) Frames after forgery [48]

d. Inserting Frame

In this type of forgeries, frames are inserted at different random points to conceal criminal activities or to create a fabricate evidence. Frames can be from the same video or any other. Figure 2.23 illustrates this forgery in a video where a forged video is created by inserting frames I1 and I2 from another film between the original video sequence's second and third frames.

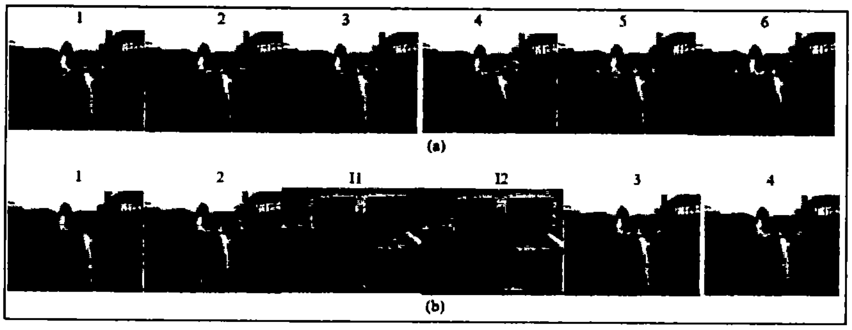


Figure 22.23. (a) Original frames (b) Video created by forging I1 and I2 frames in between 2nd and 3rd frame)[48].

e. Frame Shuffling/Replication:
This forgery rearranges or alters the original arrangement of video frames, giving the original film a new meaning. In Figure 2.24, a frame shuffling forgery in video is shown, in which certain frames in an original video sequence are shuffled, where (a) is the original video sequence and (b) is the forged video sequence where the fourth frame is shuffled with the second frame.

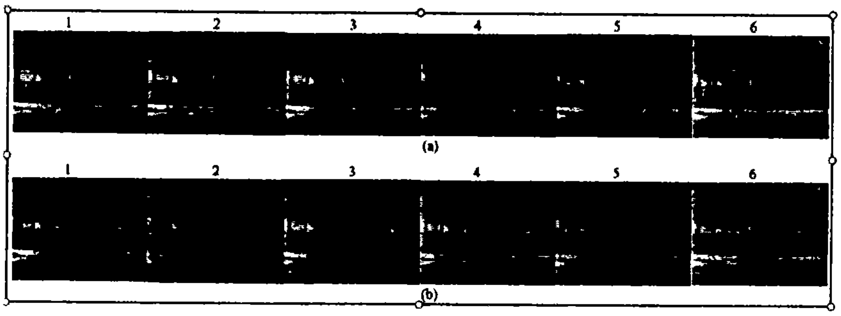


Figure 2 24· (a) The original video sequence before frame shuffling (b) Shuffling 4th frame with the 2nd frame [48].

2.11.2 Literature Review Tampered Frames Detection

Fei et al. [30] inserted a crypto hash within the video frame to protect against frame insertion and modification. S. Chen et al. [31] developed a verification technique for detecting tampering in surveillance videos by means of chaotic systems and DCT discrete cosine transform. The timing information in frames is taken as chaotic system parameters, resulting in the injection of a noise-like watermark into the domain of the block-based discrete cosine transform. A maximum likelihood estimator is used to demodulate the stored data. To detect temporal manipulation, the difference between the retrieved and experimental timing information is utilized. Spatial manipulation is discovered by comparing the extracted watermark to the original. D. Xu et al. [32] split an image watermark into numerous binary pictures and placed them directly into the compressed bit streams of several sequences in a film. This strategy was found to be successful against frame averaging attacks, scaling and filtering attacks, frame dropping and rotation attacks also.

Qianwen et al. [33] used a video forensic technique inspired by the HVS to build an automatic jump-cut detection method for analysing video alteration and tampering. They developed the 4-

EGSSIM algorithm, inspired by HVS, which enhances gradient images by the addition of logarithmic transformation. Results are evaluated in term of mean squared error (MSE) and peak signal to noise ratio (PSNR). G.H. Chen et al. [34] suggest Gradient Structural Similarity index as an alternative to the Structural Similarity index(SSIM). It uses the gradients of the input images.. Nercessian et al. [35] created the 4-SSIM and 4-GSSIM for smooth areas, edge change, texture, and edge preservation, a mix of local similarity measures and dynamic weights is used. A. Gironi et al. [36] analyzed when frames were deleted or inserted using algorithms. They used a static group of pictures (GOP) for encoding and then encoded again as frames are added or deleted. The precise location of frame insertion or removal, on the other hand, is unknown.

P. Bestagini et al. [37] detect substituted regions, an area that is replaced with a sequence of fixed images or with a chunk of the same video at varying time intervals. This strategy fails if the swapped sequence is from another video. R. Singh et al. [38] make use of both absolute and relative local correlations. Watermarking the video frames was accomplished using local relative correlations. C.-Y. Liang et al. [39] advocated for both the spatial and temporal domains tampering the shot segmentation. R.D.Singh et al. [40] detecting inter frame forgeries using pixel correlation and noise pattern analysis. M. Fallahpour et al. [41] discover spatiotemporal. Macroblocks with erroneously obtained index assist in detecting innocuous recompressions, distortions or filtering. If these macro blocks recur in the same frame and place, malicious attempts are suspected. Sensory Pattern Noise SPN based method is proposed by Hyun et al. [42] by using Minimum Average Correlation Energy (MACE) filter to detect forged regions in video. Different forgeries are identified using the scalar factor and correlation coefficient, including partial modification, video alternation, and upscale-crop. Kobayashi et al. [43] identified fabricated sections in video based on noise discrepancies. The photon shot recording device noise is used as evidence, and a linear Noise Level Function (NLF) is used to analyze the relationship between the extracted noises in order to detect forgeries. Fayyaz et al. [44] proposed an approach for detecting temporal copy-paste in painting forgery using SPN and noise residue correlation. To identify forgeries, the noise residue patterns are recovered from each video frame and compared to the acquired SPN using adaptive DCT filtering. A concise review of these techniques is given in table 2.3 below.

Table 2.4 Literature Review Tampered Frame Detection in Videos

Paper Title	Approach	Analysis
Fei et al. (2006) [30]	Inserted a MAC to detect tampering	The security of MACs relies heavily on the secrecy of the key. If the key is compromised, all communications protected by that key are vulnerable.
S. Chen et al. (2008) [31]	The disparity between timing information in recovered frames detects temporal manipulation.	Upscale and splicing methodology cannot be retrieved by timing information

D. Xu et al.(2011) [32]	Split a watermark into binary images and insert directly in the compressed bit streams of several video sequences	Watermark cannot detect upscale crop and splicing
Qianwen et al.(2017) [33]	They developed the 4-EGSSIM algorithm and used MSE and PSNR for evaluation.	No strategy for frame selection adopted
G.-H. Chen et al.(2006) [34]	Gradient Structural Similarity index was used	Gradient Structural Similarity index at regions gave positive results and can be utilized for tampering detection
S. Nercessian et al. (2011) [35]	Proposed 4 component based GSSIM	The lack of a perceptual model that can account for the human visual system's sensitivity to different spatial frequencies and orientations. The difficulty of finding optimal weights for the four components that can reflect the relative importance of each component for different types of images and distortions. The high computational complexity of the gradient component, which requires the calculation of the gradient magnitude and direction for each pixel in the image.
Gironi et al (2014). [36]	Detect frame loss and insertion.	The serial selection of pixels within the frame that is used for embedding the information inside it. This makes the method vulnerable to electronic attacks that can detect the hidden data by analyzing the pixel patterns

P. Bestagini et al. (2013)[37]	Identify region substitution.	
R. Singh et al. (2009)[38]	Use both the local and embedding local relative correlations offered a technique for locating tampering in both the geographical and temporal domains using shot segmentation.	The assumption of uniform motion between consecutive frames, which may not hold true for complex or dynamic scenes with multiple objects or camera movements. The sensitivity to noise and compression artifacts, which may affect the accuracy and reliability of the correlation coefficients and the tampering detection. The high computational cost of calculating the correlation coefficients for each frame pair and segmenting the video into shots, which may limit the applicability and scalability of the method.
C.-Y. Liang et al. (2008) [39]	Detect inter frame forgeries based on pixel correlation and noise pattern analysis.	Histogram and graph Analysis And Fmg calculation used
R.D.Singh et al. (2017)[40]	Detect inter frame forgeries by analyzing pixel correlation and noise patterns.	The assumption of uniform noise distribution across the video frames, which may not hold true for videos captured by different devices or under different conditions. The sensitivity to compression and transcoding, which may alter the pixel correlation and noise patterns and affect the detection performance. The high computational complexity of calculating the prediction residual and the noise patterns for each frame

		pair, which may limit the applicability and scalability of the method.
M. Fallahpour et al. (2014) [41]	Uses the block-based discrete cosine transform (DCT) and the singular value decomposition (SVD) to detect and locate the regions where frames have been altered in a digital video.	The assumption of block-wise uniformity of the DCT coefficients, which may not hold true for complex or textured regions in the video frames. The sensitivity to noise and compression, which may affect the accuracy and reliability of the SVD values and the tampering detection. The high computational complexity of calculating the DCT and the SVD for each block pair, which may limit the applicability and scalability of the method.
Hyuan et al. (2013)[42]	Uses the sensor pattern noise (SPN) and the minimum average correlation energy (MACE) filter to detect and locate the regions where frames have been up scaled in a digital video	The assumption of uniform scaling factor for the whole video, which may not hold true for videos that have been manipulated with different scaling factors for different regions or frames. The sensitivity to noise and compression, which may reduce the quality and reliability of the SPN extraction and the MACE filter matching. The high computational complexity of calculating the SPN and the MACE filter for each frame, which may limit the applicability and scalability of the method.

Kobayashi et al. (2009)[43]	The paper uses a video forensic method that uses the photo response non-uniformity (PRNU) noise, also known as the sensor pattern noise (SPN), to detect and locate the regions where frames have been copied or moved in a digital video	The assumption of uniform PRNU noise across the video frames, which may not hold true for videos captured by different devices or under different conditions.
Fayyaz et al. (2020)[44]	The paper uses singular point network (SPN) to extract the noise residue from each frame of a video and then computes the noise correlation between adjacent frames to detect the copy-paste forgery1. It also uses the noise correlation with the previous frames to identify the forged frames and locate the tampered regions in the video2.	The paper does not provide any comparison with existing methods for copy-paste forgery detection, such as SIFT1, SURF2, or ORB3. It is unclear how the proposed method performs against these methods in terms of accuracy, robustness, and efficiency. The paper does not provide any analysis of the impact of different parameters on the performance of the proposed method, such as the block size, the threshold value, and the noise correlation function. It is not clear how these parameters are chosen and how they affect the detection results. The paper does not address the issue of anti-forensics, which is the process of concealing or removing traces of forgery from an image. For example, an attacker could apply post-processing techniques such as JPEG compression, blurring, or noise addition to reduce the detectability of copy-paste forgeries. It is not clear how

		the proposed method would cope with such scenarios and what countermeasures could be adopted.

2.12 Problem Statement

The most well-known techniques for video steganography include modifying the LSB of the carrier, for a picture or a digital video frame. Lately, DNA-based solutions have developed, although they need the supplement of any DNA thread xored with a user-supplied key. Firstly, this random strand of DNA used as a key [4] is the major cause of insecurity. There are various threats that can compromise a key's security. Here are some of the major threats to consider:

1. Weak keys: Using keys that are short or predictable increases the likelihood of them being cracked. Strong keys should be generated using high-quality random number generators.

2. Incorrect use of keys: Keys should be used specifically for their intended purpose and algorithm. Using a key for a different purpose may weaken its protection.

3. Reuse of keys: Reusing keys across different systems or encryption instances can make it easier for attackers to exploit any vulnerabilities and compromise the key's security.

4. Non-rotation of keys: If a key is used excessively without being rotated or renewed, it becomes more susceptible to attacks. Regular key rotation helps mitigate the risks associated with prolonged key usage.

By being aware of these threats and implementing proper key management practices, such as using strong and unique keys, adhering to key usage guidelines, avoiding key reuse, and regularly rotating keys, the security of encrypted data can be significantly enhanced. There have been several scientific research studies that suggest using a single key for encryption or steganography is not secure. If attacked, this might readily result in the decoding of the encrypted text.

Sun et al [68] analyzed the security of the RC4 encryption algorithm against key exhaustion attacks and showed that an attacker can easily recover the plaintext if the same key is used repeatedly. In another research by Smith et al [69] a comprehensive analysis of key exhaustion attacks on symmetric encryption algorithms has been done that highlights the vulnerabilities of using a single key for encryption. Alice Johnson et al [70] have also explored the weaknesses of single-key steganography algorithms and demonstrated how an attacker can detect and extract the hidden information by analyzing the encrypted data.

These research papers provide evidence and insights into the security risks associated with using a single key for encryption or steganography. They emphasize the importance of employing stronger encryption techniques and key management practices to ensure the confidentiality and integrity of sensitive information.

Without secure steganography techniques, protected data concealed within videos can simply be tampered with or removed. Video tampering, which entails the addition of new frames, the deletion of existing frames, or the altering of critical data, has received diminutive consideration for a steganographic video. In a steganographic video, any mess around might result in the cypher text being corrupted and vital information being lost. Following the literature review and discussion above, the following problems were observed in the realm of video steganography.

a. Problem I

Exhausting a single DNA sequence through the whole encryption process compromises cypher text's security. If this DNA structure is compromised, the encryption text is easily deciphered.

b. Problem II

After encryption process, the cypher text is inserted in a video for concealing. Embedding text in a video requires selection of suitable frames. Without inspecting the contents of the frame, concealing information in an inappropriate frame can generate new artefacts into the video that can aid an attacker in determining the message placements.

c. Problem III

Detecting forgeries in steganography video has been a demanding area of research. Upscaling and splicing forgeries have been discovered in videos by utilizing pixel correlations and noise fluctuations. No such work is visible for detecting such forgeries in steganography videos.

2.13 Research Questions/Open Issues

Under the light of the literature survey the following questions are yet not answered and our goal is to provide a solution to them.

1. How much security of data can be improved by utilizing DNA for embedding rather than being used as a key for selection or encryption?

2. Can selection of a better set of frame/frames to hide the DNA based secret message in a video improve the security of the cipher text and keep imperceptibility of the video intact.

3. How to identify insertion, deletion and inter frame forgeries, in a video with an embedded cipher text?

2.14 Summary

Based on the comprehensive literature survey conducted in the field of video steganography, it is evident that significant work has been accomplished in terms of understanding the terminology, techniques, encryption methods, and potential areas for further research. However, there is always scope for improvement in any field, including the existing techniques employed in video steganography.

After conducting a thorough literature review, it is evident that there are two key areas in video steganography that require improvement to maintain imperceptibility. The first aspect is the utilization of advanced or unbreakable encryption schemes. The existing encryption methods may have vulnerabilities or limitations that can potentially compromise the security of the hidden information. Therefore, adopting more advanced encryption techniques is crucial to enhance the overall security of video steganography.

The second area that necessitates improvement is the process of frame selection for embedding. It is essential to ensure that the selected frames do not introduce visible artifacts or distortions that could raise suspicion or compromise the covert nature of the embedded information. Current frame selection methods may not be foolproof and can lead to the detection of the hidden data. Thus, more robust approaches are needed to ensure that the embedded information remains imperceptible within the video.

By addressing these two aspects, namely advancing the encryption schemes and refining the frame selection process, the imperceptibility and security of video steganography can be significantly enhanced. This research gap provides an opportunity for further exploration and the development of novel techniques to overcome the existing limitations in video steganography. In light of these findings, the upcoming chapter proposes a solution that aims to enhance video steganography by incorporating DNA-based techniques and intelligent frame selection. This approach is expected to address the limitations identified in the existing methods and contribute to the overall improvement of video steganography techniques.

By leveraging the unique properties of DNA and employing intelligent frame selection algorithms, the proposed solution seeks to enhance the security, imperceptibility, and reliability of video steganography. It is anticipated that this research will pave the way for advancements in the field and open up new avenues for further exploration and innovation.

Chapter 3

Secure Data Hiding Scheme in Complex Frames in Video Steganography

3. Introduction

This chapter discusses the proposed solution in detail. It begins with the introduction of frame selection and series of work already done with respect to frame selection. It introduces the proposed approach along with flow charts. It discusses the proposed solution in a serial manner along with new algorithms, flowcharts and images. The chapter concludes with the discussion on experimental results.

3.1 Introduction to the Proposed Approach

A video is made up of a collection of images. Every individual image in this collection is called a frame. When these frames are played one by one at a certain speed for example 24 frames per second, the human visual system perceives these frames as in motion hence creating a video. Frame Selection in video steganography is the process of selecting a subset of frames from the complete set of frames forming a video. This is a very important step in video steganography as the secret text, image or audio is embedded in the subset of frames.

In video steganography, the data is subjected to a series of preprocessing steps before embedding. These steps may can involve various types of preprocessing, encryption or transformations. It all depends on the preprocessing algorithm devised by the researcher that either he chooses one of them or a combination of these. The ultimate goal is to generate a secure, undetectable form of original data. The generation of secure algorithm is an active field of research wherein we have along with frame selection also devised an algorithm. A common approach is that data in the video is embedded in any pseudo randomly selected frames of a video. In such algorithms the contents of the frames are not taken into consideration as the frames are selected using pseudorandom selection algorithms. The selected frames might have very low intensity pixels such that any minor change in intensity value due to the embedding can clearly show the difference between the original frame and the stego frame. This way that embedded data is perceptible in the stego video hence giving first chances of detection to any hacker. Similarly, a frame with high intensity values can prove to be a good candidate for embedding in several ways dependent upon the embedding algorithm.

The inspiration for this research was the lack of attention in the selection of good frames for embedding data. For example Nirmalya et al. [4], Ramalingam et al. [5] and Kelash et al. [8] embed in random frames without looking in to the contents of the frame this approach introduces artifacts in the frame that make the video itself perceptible.

Cao et al. [6], Bin et al. [7] and Shuvendu et al. [29] embedded data in motion vectors. Embedding in motion vectors has high computational because it has to search for the best possible velocity vector out of an N × N set of possible velocity vectors. This technique

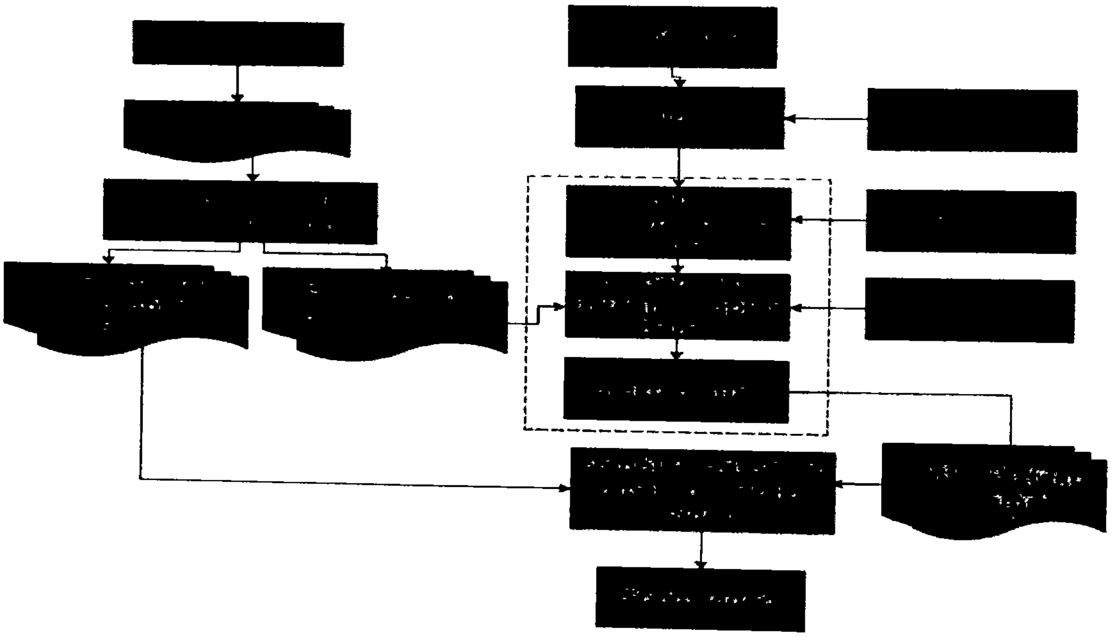


Figure 3.1. Flow chart of the proposed DNA scheme

3.2. Frame selection

We will divide the cover medium video into multiple frames and then choose few frames to work with. The frame selection module is a key element of our suggested solution for this. In this module, we, segregate complex frames of our video. Complex frames are those frames that involve any scene change. We will embed our data only in these frames. The reason behind using only complex frames for this purpose is to camouflage the secret data behind the motion estimation information. Frames involving scene change have object movement, color change, object infusion, and diffusion. So, taking advantage of this information secret text is infused only in these frames. For the detection of complex frames, we have used the key frame extraction technique mentioned in[59]. Discrete Cosine Transform i.e. DCT coefficients are extracted from each frame of the video. Using the difference of the Mean of two consecutive frames, frames with scene change are segregated. If the mean of consecutive frames is not the same, it means there is some change between these frames. Hence that frame is selected as a complex frame. This is done for all frames of the video to achieve a subset of frames known as the complex frames. This procedure is given in the form of the algorithm in Table 3.1 below. After the selection of complex frames, the next step is treating the text to be embedded. We name this step data encryption.

Table 3.1 Frame Selection Algorithm

Algorithm 1: Frame Selection Algorithm

```
Frame Selection Algorithm:
Input: Video P

Output: An array of complex frames
    1.    for each frame i
    2.    find DCT of frame i
    3.    find MEAN of the DCT of each frame
    4.    if MEAN of the frame(i) and frame(i - 1) is not the same select the ith frame as a complex frame
    5.    selected frame(ta)= frame(i)
    6.    end
7.    end
```

3.2.1 Time Complexity

The time complexity of an algorithm is a measure of how the running time of the algorithm increases as the size of the input increases. It helps us understand the efficiency of an algorithm in terms of the time required to execute it. Overall, the complexity of the entire algorithm depends on the complexities of the functions and can vary depending on the specific details of their implementations.

1. Time Complexity of Proposed Approach

The complexity of the algorithm in Table 3.1 can be analyzed by considering the operations performed within each step and the number of iterations involved.

1. The first step iterates over each frame, resulting in a complexity of O(N), where N is the number of frames in the video.
2. Finding the DCT (Discrete Cosine Transform) of a frame typically has a complexity of O(M^2), where M represents the size of the frame.
3. Calculating the mean of the DCT coefficients involves summing up all the coefficients and dividing by the total number of coefficients. This operation has a complexity of O(M^2).
4. Comparing the mean of the current frame with the mean of the previous frame requires constant time, resulting in a complexity of O(1).
5. Assigning the selected frame to the "selected frame" variable is a constant time operation, resulting in a complexity of O(1).
6. Ending the current iteration is also a constant time operation, resulting in a complexity of O(1).
7. The second "end" marks the end of the loop iterating over each frame and is also a constant time operation, resulting in a complexity of O(1).

Considering all these steps, the overall complexity of the algorithm can be approximated as O(N * M^2), where N is the number of frames and M represents the size of each frame.

2. Time Complexity of Algorithm [4]:

To determine the complexity of the algorithm, we can analyze each function of the algorithm [4] separately: Convert the DNA strand to binary representation and determine the length of the binary sequence. The time complexity of this step depends on the length of the DNA strand and can be considered as O(P), where P is the length of the binary sequence.

1. Count the number of codons in the DNA strand. The time complexity of this step depends on the length of the DNA strand and can be considered as O(a), where a is the count of codons.
2. Convert the DNA strand to its corresponding decimal values. The time complexity of this step depends on the length of the DNA strand and can be considered as O(Xn), where X is the length of the random number and n is the length of the DNA strand.
3. Use the linear congruential generator to generate a pseudorandom number by combining the decimal values, codon count, number of digits in decimal values, and the key. The time complexity of this step depends on the lengths of the inputs and can be considered as O(X + a + m).
4. Compare the length of the random number with the length of the DNA strand and apply DNA cutting or polymerization properties accordingly. The time complexity of this step depends on the lengths of the random number and DNA strand and can be considered as O(Q), where Q is the length of the DNA strand.
5. Create a replica of the key DNA strand and compare it with the random number until both have the same length. The time complexity of this step depends on the lengths of the key and random number and can be considered as O(V), where V is the length of the replica.
6. Perform the XOR operation between the key and the random number. The time complexity of this step depends on the length of the key and can be considered as O(m).
7. Begin the key generation process using the result of the XOR operation. The time complexity of this step depends on the length of the key and can be considered as O(N).
8. Generate a chaos sequence for each frame in M*N dimensions. The time complexity of this step depends on the size of the frame and can be considered as O(M * N).
9. Divide the encrypted text of length n among each frame subset of size M * N. The time complexity of this step depends on the size of the subset and can be considered as O(n / (M * N)).

Overall, the complexity of the algorithm can be approximated as O(P + a + Xn + X + a + m + Q + V + m + N + M * N + n / (M * N)), but it may vary depending on the specific implementation details and additional operations not mentioned in the given description.

3 Comparison of Time Complexities:

When an algorithm has a Big O complexity, it represents the upper bound of the algorithm's worst-case time complexity. In other words, it provides an estimate of how the algorithm's runtime will grow relative to the input size as the input size approaches infinity.

The Big O notation expresses the growth rate of the algorithm's runtime as a function of the input size, usually denoted as "n." For example, if an algorithm has a complexity of O(n), it means that the algorithm's runtime will increase linearly with the input size. If the input size doubles, the runtime will roughly double as well.

The Big O complexity provides an approximation of the algorithm's scalability and efficiency. It allows us to compare different algorithms and understand how their performances may differ for large input sizes. Algorithms with lower Big O complexities are generally more efficient because their runtimes grow at a slower rate compared to algorithms with higher complexities.

It's important to note that the Big O notation only represents the upper bound and worst-case scenario of an algorithm's complexity. It doesn't provide information about the best-case or average-case performance. Additionally, the Big O notation doesn't capture constant factors or lower-order terms, which means two algorithms with the same Big O complexity can still have different actual runtimes. Therefore, Big O complexity should be used as a rough estimation and a tool for algorithmic analysis and comparison.

When comparing the complexities of two algorithms, O(N * M^2) and O(P + a + Xn + X + a + m + Q + V + m + N + M * N + n / (M * N)), the notation O(N * M^2) signifies that the first algorithm's complexity grows quadratically with the size of parameter M and linearly with the size of parameter N. This indicates that the runtime of the algorithm increases significantly as the values of M and N increase.

On the other hand, the notation O(P + a + Xn + X + a + m + Q + V + m + N + M * N + n / (M * N)) represents the second algorithm's complexity as a sum of several terms. Each term represents a specific operation or input size that contributes to the overall runtime of the algorithm. The complexity analysis suggests that the runtime of the second algorithm is influenced by the sizes of parameters P, a, X, n, M, N, Q, and V, as well as the constant terms m.

Comparing the two complexities, it is evident that the first algorithm's growth rate is dominated by the term N * M^2, while the second algorithm's growth rate is determined by the sum of various terms.

3.3. Data encryption

Data encryption involves two phases of encryption. The first one is the encryption of the text message by a cipher. After this encryption, we add another layer of security by using the concept of DNA Cryptography. We embed our encrypted text in DNA using the technique referred to by [4]. We call this data encryption algorithm and is explained in table 3.2. The algorithm starts with the encrypted message ***m*** converted to binary form M. The message ***m*** will be the encrypted message that the sender chooses to transmit to another client who is located on a different network. Any cipher can be used to convert any plaint text to ***m***. We have used AES cipher in our experiment. The description of AES cipher has already been given in chapter 1.

The algorithm comprises of three sub-phases. At the last stage of the algorithm M is converted to M''' i.e the newly created fake DNA. The first sub-phase converts M to M′ by implementing the DNA base pairing rules. M′ contains nucleotides sequences. By applying the DNA base pairing rules, the message can be converted from binary to DNA sequence. Not only DNA base pairing helps to encrypt the message from binary to DNA sequence but also it is applied to decrypt the secret message.

Let's demonstrate the first phase by a step-by-step example. Let **M=100111000011** be the binary converted form of the encrypted message ***m***. Let **$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}$ CC_{20}** be the **DNA** reference sequence.

DNA Sequence:

$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}$ CC_{20}

M: 100111000011

Sub-phase1: Let A= 00, T= 01, C= 10, G= 11, then M′ will be M′= CTGAAG

The subsequent (second) sub-phase entails implementing the supplementary rules. This step is intended to increase the difficulty of retrieving the initial plain text message. By applying the complementary principles, a new form of M', known as the M" is achieved.

Sub-phashe2: ((AC) (CG) (GT) (TA)), then M″ will be M″= GATCCT.

The sender and receiver will chose the same DNA reference sequence from a large number of possibilities provided by European Bioinformatics Institute i.e. EBI, the international repository for open data in the life sciences or the National Center for Biotechnology Information advances science and health i.e. The NCBI database by providing access to biomedical and genomic information. The exact role of the third sub-phase is, extracting the index of each couple nucleotides in DNA reference sequence, numerically. When all the indexes have been extracted for each couple nucleotide, M‴ is created as in the example below.

DNA Sequence:

$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}$
CC_{20}

Sub-phase3 (Indexes): M‴= 8137

M‴ is precisely the secret message with some changes through the embedding phase. Now this M''' can be embedded in any video frame pixel. This algorithm is shown in figure 3.3. Once the secret message is ready, the next step is its embedding in the video spread among the complex frames.

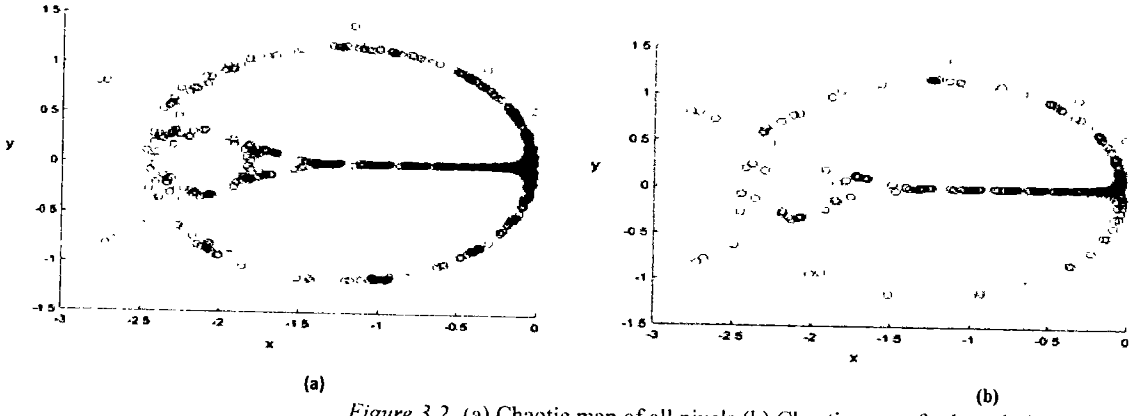


Figure 3.2. (a) Chaotic map of all pixels (b) Chaotic map of selected pixels

To add further security and maintain video perceptibility neither all frames nor are all pixels will be utilized of these complex frames. This has been done by selecting pseudo-random pixels using Burger chaotic map and linear congruential generator [4]. As chaotic maps seed value always changes for each

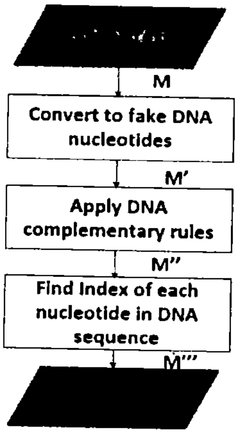

M
Convert to fake DNA nucleotides
M'
Apply DNA complementary rules
M"
Find Index of each nucleotide in DNA sequence
M'''

Figure 3.3. Conversion of cipher text to DNA code

frame therefore the generated pixel set will always change making it unidentifiable for an attacker. Figure 3.2 shows the x-y plot of burger chaotic map for the pixels generated by the burger chaotic map and those selected for embedding. The pixel set generated by the chaotic map is fed to the linear congruential generator that further adds randomness to the selected pixels by reducing the pixel set further. Because of its relative ease of use, linear congruential generators are one of the most seasoned and well-known systems for producing random numbers [4]. The permutation cypher is used to embed 3 bits from each of red and green, and 2 from blue as proposed in [9] further maintaining video perceptibility. Figure 3.7, a and b shows the projection of chaotic map of selected pixels in rhinos and the projection of its sub pixels on the image frame.

Table 3.2 Data Encryption Algorithm

Algorithm 2: Data Encryption Algorithm

Data Encryption Algorithm:

```
Input: Cipher Text M
Output: Fake DNA
Convert M to binary
for each binary pair $X_i$ in M’
    Apply DNA complementary rule such that replace base A with C, base C with G, base G with T and base T with A to get M’’
end
for each pair $X''_i$ in M’’
    extract the index of each couple nucleotide using the reference DNA numerically
end
```

In the mathematical model for the proposed system let M represent the cipher text. Let M' represent the binary conversion of M. Let M'' represent the transformed sequence from M' after applying the DNA complementary rule. Let I be the set of indices of each nucleotide couple in M'' using the reference DNA numerically. The above algorithm for converting the encrypted text M to M''' i.e. the fake DNA indices, can be expressed mathematically as follows:

1. Function ConvertToBinary(M): This function takes as input the cipher text M and returns the binary representation, M'. Mathematically, it can be represented as:

M' = ConvertToBinary(M)

2. Function ApplyComplementRule(M'): This function takes as input the binary representation M' and applies the DNA complementary rule (A ↔ C, G ↔ T) to each binary pair in M' to get M''. Mathematically, it can be represented as:

M'' = ApplyComplementRule(M')

3. Function ExtractIndices(M")**: This function takes as input M" and extracts the index of each nucleotide couple using the reference DNA numerically to get the set I. Mathematically, it can be represented as:

I = ExtractIndices(M")

Therefore, the entire algorithm can be summarized in the following mathematical model:

Algorithm FakeDNAGeneration(M)

1. M' = ConvertToBinary(M)

2. M" = ApplyComplementRule(M')

3. I = ExtractIndices(M")

Where:

- ConvertToBinary(M) represents the binary conversion of the cipher text M.

- ApplyComplementRule(M') represents the application of the DNA complementary rule to M'.

- ExtractIndices(M") represents the extraction of the index of each nucleotide couple in M".

The output of the algorithm, the fake DNA, is represented by the set I.

3.3.1 Enhancing Security through Frame Selection with Scene Changes in Video Steganography

The literature survey 2.1 highlights the crucial role of carefully selecting frames to ensure efficient concealment of embedded information. Neglecting this aspect can lead to steganography videos being susceptible to eavesdropping, as visible artifacts may compromise the secrecy of the hidden data. Several studies have examined the effectiveness of incorporating scene changes as a means to enhance the security of concealed information. Here are some ways in which efficient frame selection helps in enhancing security:

Increased imperceptibility: Selecting frames that closely match the statistical characteristics of the video content improves the imperceptibility of the embedded information. Frames with similar visual features, such as color distribution, texture, and motion, can effectively hide the embedded data within the natural variations of the video. This makes it harder for unauthorized users to detect the presence of hidden information.

Reduced detectability: Frames with complex content, such as scenes with significant motion or visual variations, provide better cover for the embedded data. These frames introduce additional noise and variations, making it challenging for steganalysis techniques to detect the hidden information. Efficient frame selection considers these factors and strategically embeds data in frames where the changes caused by the embedding process are less likely to be detected.

Robustness against attacks: Effective frame selection can enhance the robustness of the steganographic system against various attacks. By choosing frames with high redundancy or less sensitive content, the hidden information becomes more resistant to compression, transcoding, or other post-processing operations that may introduce artifacts or distortions. Robust frame selection ensures that the embedded data remains intact and retrievable even under different attack scenarios.

Reduced capacity limitations: Video steganography often faces limitations in terms of the amount of data that can be embedded within the video while maintaining its quality and imperceptibility. Efficient frame selection allows for optimal utilization of available frames, ensuring that the embedded information is distributed effectively across the video. This helps maximize the capacity for data embedding while minimizing the risk of overloading specific frames and increasing the chances of detection.

Increased computational efficiency: Selecting frames strategically can also lead to improved computational efficiency. By focusing on specific frames that offer suitable characteristics for embedding, the processing time and resources required for data embedding can be optimized. This enables faster and more efficient steganographic operations, making the system more practical for real-time or large-scale applications.

Overall, efficient frame selection in video steganography not only improves the imperceptibility and robustness of the hidden information but also enhances the overall security of the steganographic system by reducing detectability, increasing resistance to attacks, and optimizing data capacity and computational efficiency.

3.4. Extraction Module

The original message from the stego movie is extracted using the data extraction procedure as seen in figure 3.4 and algorithm in table 3.3. For this, the frames of the stego video will be separated again. This module uses the same parameters as in the encryption phase. For the data extraction separate the stego video into the same number of frames and under the same parameters as in the time of encryption. The frame numbers, pixel values, AES secret key have been shared between two

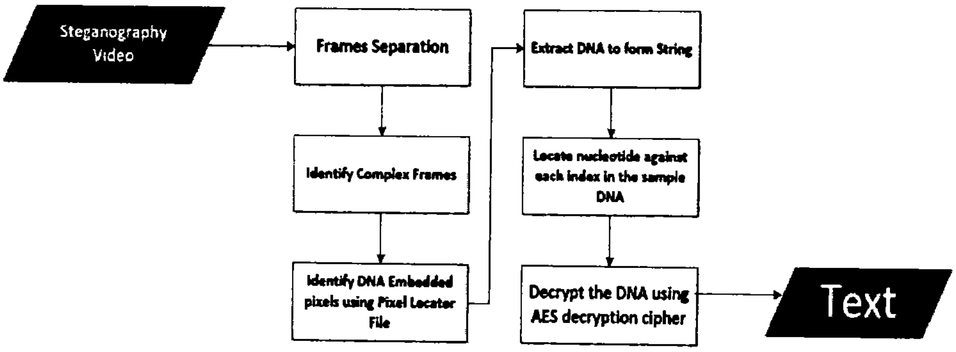

Steganography Video
Frames Separation
Identify Complex Frames
Identify DNA Embedded pixels using Pixel Locater File
Extract DNA to form String
Locate nucleotide against each index in the sample DNA
Decrypt the DNA using AES decryption cipher
Text

Figure 3 4· Extraction of cipher text from DNA.

parties in an encrypted form. Locate the complex frames F in which you have saved your encrypted content. Locate the embedded pixels within the stego frames using the pixel points given by the chaotic map and shared. Each pixel's bits should be retrieved in the order 3, 3, 2, i.e. 3 bits each from red and green, and 2 from blue. The values retrieved are the index values in the reference DNA i.e. M and 2 from blue. The values retrieved are the index values in the reference DNA i.e. M``` i.e **M‴=** i.e **M‴= 8137.**

DNA Sequence:

AT$_1$CG$_2$AA$_3$TT$_4$CG$_5$CG$_6$CT$_7$GA$_8$GT$_9$CA$_{10}$CA$_{11}$AT$_{12}$TC$_{13}$GC$_{14}$GC$_{15}$TG$_{16}$AG$_{17}$TG$_{18}$AA$_{19}$ CC$_{20}$

M‴= 8137

M″= GATCCT

In the reference DNA, replace each index with its nucleotide pair this will give us the M``. Then M`` is converted to M` by applying DNA inverse of complementary rules.

Sub-phashe2: ((CA) (GC) (TG) (AT)), then M′ will be M′= CTGAAG

M is converted to binary form by replacing a binary equivalent with each protein. For this each protein will be replaced with the same binary equivalent as was used for encryption. Replace **A= 00, T= 01, C= 10, G= 11, then M′ will be M′= CTGAAG** Finally decrypt M using AES encryption to reveal the hidden text in plain text form.

Table 3.3 Data Decryption Algorithm

Algorithm 2: Data Decryption Algorithm

```
Data Decryption Algorithm:
Input: Stego Video
Output: Original secret message in plain text form

1. Decrypt the shared file containing key, selected frames and pixel information
2. Identify the complex frames and pixel values using the decrypted file
3 for each embedded pixel X_i in Selected Frame F_i
4. Retrieve pixel's bits in the order 3, 3, 2 i.e. 3 bits each from red and green, and 2 from blue
```

```
    Concatenate the retrieved bits in M```
end
for each X_i in M```
    Replace Xi with its matching nucleotide couple and concatenate the retrieved nucleotide pairs in M``
end
for each binary pair X_i in M``
    apply DNA complementary rule such that replace base C with A, base G with C, base T with G and base A with T to get M`
end
Let A= 00, T= 01, C= 10, G= 11
Replace each nucleotide pair in M` with its binary equivalent to generate a binary number M
Decrypting M with the decrypted key and cipher provides the original secret message in text form
```

3.5. Results and Discussions

We have experimented with this scenario with multiple videos of different lengths. The detail of these videos is given bellow. For experimentation dataset used is the same as in [4]. Results are discussed for the sky.mpeg4 file that has a duration of 14 seconds and a file size of 2.91 Mb. In the input video file, there are 441 frames in total. Figure 3.5 also shows the results of the proposed approach in some of the other videos. It shows both the original and stego images of frame 70 of rhino video. We can see that the proposed approach maintains visual perceptibility of the frames by embedding data only in complex frames. We can see the same result in the original and stego images of frame 70 (c, d) of forest (e, f) sky and (g, h) school. To explain a sample scenario we will be using the sky.mpeg4 file as our cover video. It has a duration of 14 seconds and a file size of 2.91 Mb. In the input video file, there are 441 frames in total.

The algorithm requires 2 security keys. One is for the encryption of data using the cipher and the other is a random DNA for embedding the encrypted text. Part of the DNA strand used is "GATCACAGGTCTATCACACCCT ATTACCACT". The text has been encrypted using AES. An example of the proposed method is shown below. Xn = 2118, a = 21, c = 62, and m = 1000 are the starting values for LCG. A series of random values will be generated by LGC i.e. the linear congruential generator using the seed value 10, P = 408, Q = 408. Starting with the fourth codon, a total of 204 codons are chosen. The codon for this experiment is GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGGGTGTGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTGCCTCATTCTATTATTTATCGCACCTACGTTCAATATTACAGGCGAACATACCTACTAAAGT. The key for the AES cipher is = 12481632641282754, frame number = 7, and first value of chaotic map = 0.3. We have analyzed our method using Mean Squared Error, Peak Signal to Noise Ratio, histogram analysis and the quality of the steganographic video frames. Three videos of the same resolution and frame rate as in [2] were used as a comparison. Their detail is given in table 3.4 below.

Table 3.4 Cover Video Details

File Mpeg 4	Duration (sec)	Size (MB)ᵃ	Total Frames	Selected Frame
Forest	42	3.64	643	520
rhino	7	0.29	114	88
school	26	1.08	787	101
sky	14	2.91	447	31

3.5.1. Mean squared error (MSE)

Aggregating the differences between the corresponding pixel values of the original and the stego video frame mean square error is calculated. We calculate the difference between the original and the stego video frame, and divide the total by the frame's size using formula, $MSE_e = \sum_{n=1}^{h}\sum_{n=1}^{h}(C_{ij}^{c} - s_{ij}^{c})$. Two bytes are selected of the color c at the (i, j) location from the original frame and stego frame here in called the $C_{ij}{}^{c}$ *and* $S_{ij}{}^{c}$ respectively, c is the color component, w = width of image, h = height of image. Table 3.5 shows the comparison of the MSE values for the proposed algorithm and random frame selection technique [4] and the random DNA Algorithm [20]. The suggested approach has a lower MSE and a larger payload than the modified LSB technique [20] and the random DNA Algorithm [4].

Table 3.5 Results for MSE

Name of video file	Algorithm[20] MSE	Algorithm[4] MSE	Proposed Mechanism MSE
Rhino	13.5100000	0.0001818	**0.0001653**
Forest	16.5300000	0.0001748	**0.0001205**
Sky	11.3900000	0.0000282	**0.0000176**
School	26.5700000	0.0001375	**0.0001010**

The Fig 3.5 shows that the suggested approach gives improved result for the proposed approach as it has a lower MSE value than the algorithm [4] and algorithm[20].

3.5.2. Peak Signal to Noise Ratio (PSNR)

The quality of an image corrupted due to noise and blur can be measured using PSNR. It is used to determine the degree of similarity between the original and stego frames by calculating the difference between them. Higher the value of PSNR, indicates higher the quality rate and more similarity between the two frames. MSE is indirectly proportional to the PSNR. PSNR is computed by using the following formula $PSNR = 10\log_{10} Maxl^2 / MSE$. Where Maxl denotes the highest pixel value of a RGB image, (8 bits for each making one pixel is equal to 24 bits). PSNR is calculated separately for each of the three channels, resulting in a maximum value of (28-1) = 255. The suggested algorithm's results were compared to Video Steganography employing random frame selection techniques [5]. Same videos were chosen as in [3] with the same resolution and frame rate. Table 3.4 lists the video files and their descriptions. We may conclude that the suggested approach produced superior overall results, as shown in table 3.6, because greater PSNR indicates a smaller difference between the original and stego videos.

The suggested algorithm's results were compared to Video Steganography employing random frame selection techniques [4]. We created four videos with the same resolution and frame rate as for comparison as in [4]. Tab 3 lists the video files and their descriptions. The PSNR of the proposed approach, algorithm [4], algorithm [8] and algorithm [20] are compared in Table 3.6. We may conclude that the suggested approach produced improved results, as shown in Table 3.6.

Table 3 6 Results from DNA frame selection technique [1] and proposed algorithm for PSNR

Name of video file	Algorithm[20] PSNR	Algorithm[4] PSNR	Algorithm[6] PSNR	Proposed Mechanism PSNR
Rhino	**36.824**	32.195	32.625	**33.2126**
Forest	**33.8256**	25.3075	30.519	**32.3982**
Sky	37.793	36.4644	36.118	**42.1388**
School	34.712	31.1546	32.951	**35.8532**

The PSNR values obtained are higher than those obtained using random frame selection technique. The higher the PSNR value, the better the quality of the stego video. As a result, the proposed technique provides increased security and can produce a video that is as identical to the original cover-video as possible with minimal distortion. The proposed approach shows improved results in terms of payload as well, as shown by the payload comparison of proposed approach, algorithm [4], algorithm [6] and algorithm [20] in Table 3.7 and figure 3.6.

Table 3 7 Results for Payload

Name of video file	Algorithm[4] Payload	Algorithm[6] Payload	Algorithm[20] Payload	Proposed Mechanism Payload
Rhino	1.0174	1	0.000013	**1.1393**
Forest	1.0087	1	0.0000058	**1.0696**
Sky	1.0016	1	0.0000011	**1.0131**
School	1.0087	1	0.0000058	**1.0696**

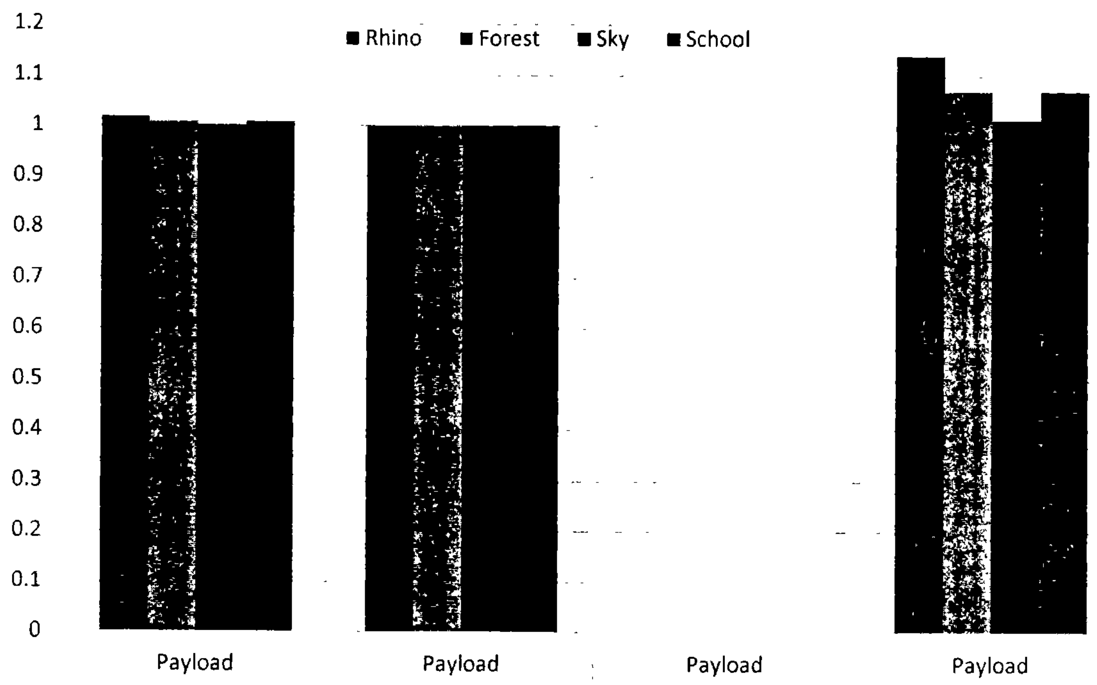
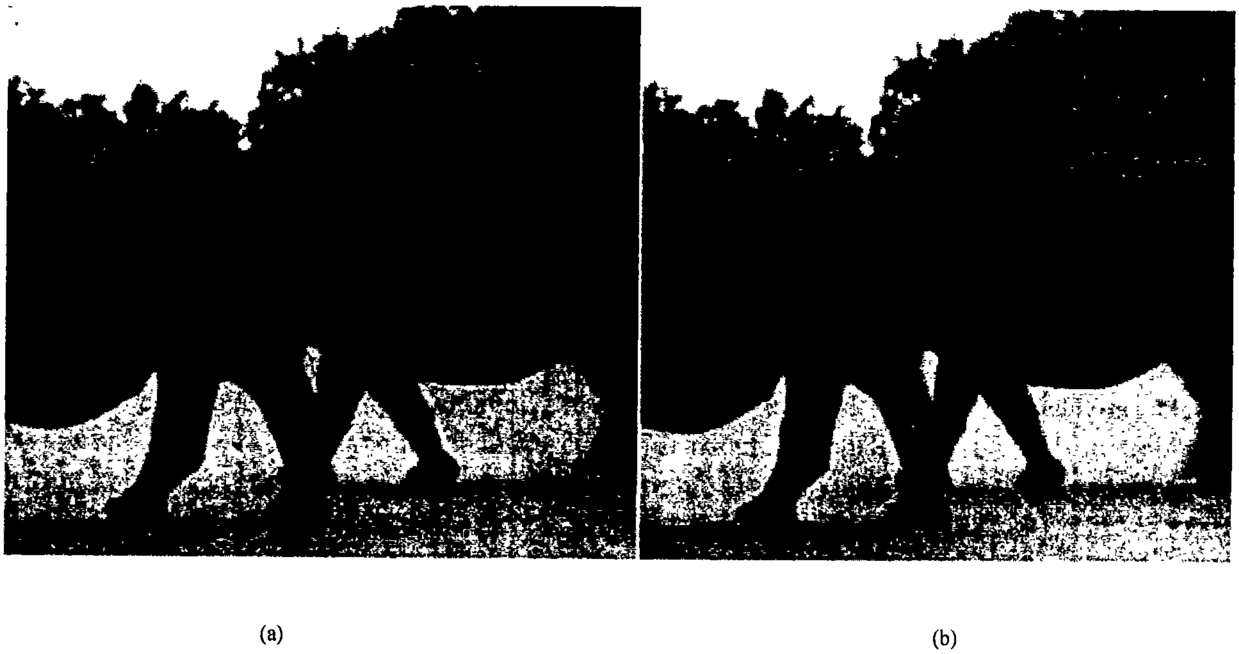


Figure 3.6: Results for payload.

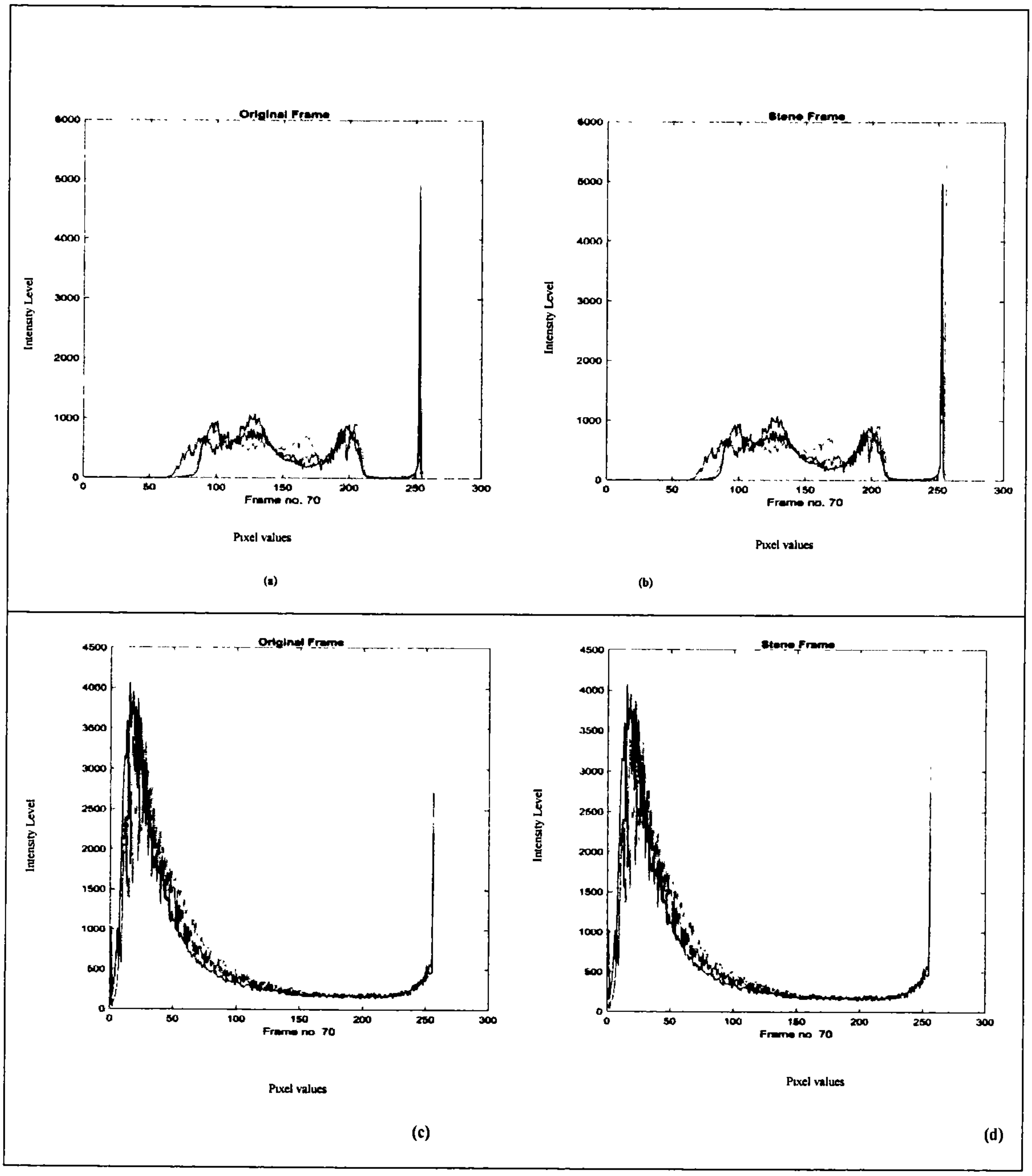
3.5.3. Histogram Difference

A histogram difference is a measure that depicts the difference between two images as a graphical representation. The histogram plots the frequency components in an image along horizontal and vertical axis. The horizontal axis indicates the pixel's RGB color level while the vertical axis indicates the how many times each color value appeared in the frame. We can see the histogram of each frame will display the amount of difference between the two frames and also give us an understanding of the video quality. Figure. 3.8, shows the histogram of selected frames of the sample stego videos rhinos, forest, and sky and their and their originals. The histogram of the frame where the scene changed abruptly, as opposed to the histogram of the frames where the scene changed gradually in all the stego frames, shows abrupt changes. The histograms show extremely mild changes due to distributed embedding in random locations among selected frames. Due to which, stego video frame alterations relative to the original video frame will neither disclose nor reduce visual imperceptibility.



(a) (b)

Figure 3.5 (a,b) Chaotic map of selected pixels rhinos and sub pixels image.


Original Frame
Intensity Level
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
Frame no. 70
Pixel values
(a)
Stego Frame
Intensity Level
Frame no. 70
Pixel values
(b)
Original Frame
Intensity Level
4500
4000
3500
3000
2500
2000
1500
1000
500
0
Frame no 70
Pixel values
(c)
Stego Frame
Intensity Level
Frame no 70
Pixel values
(d)

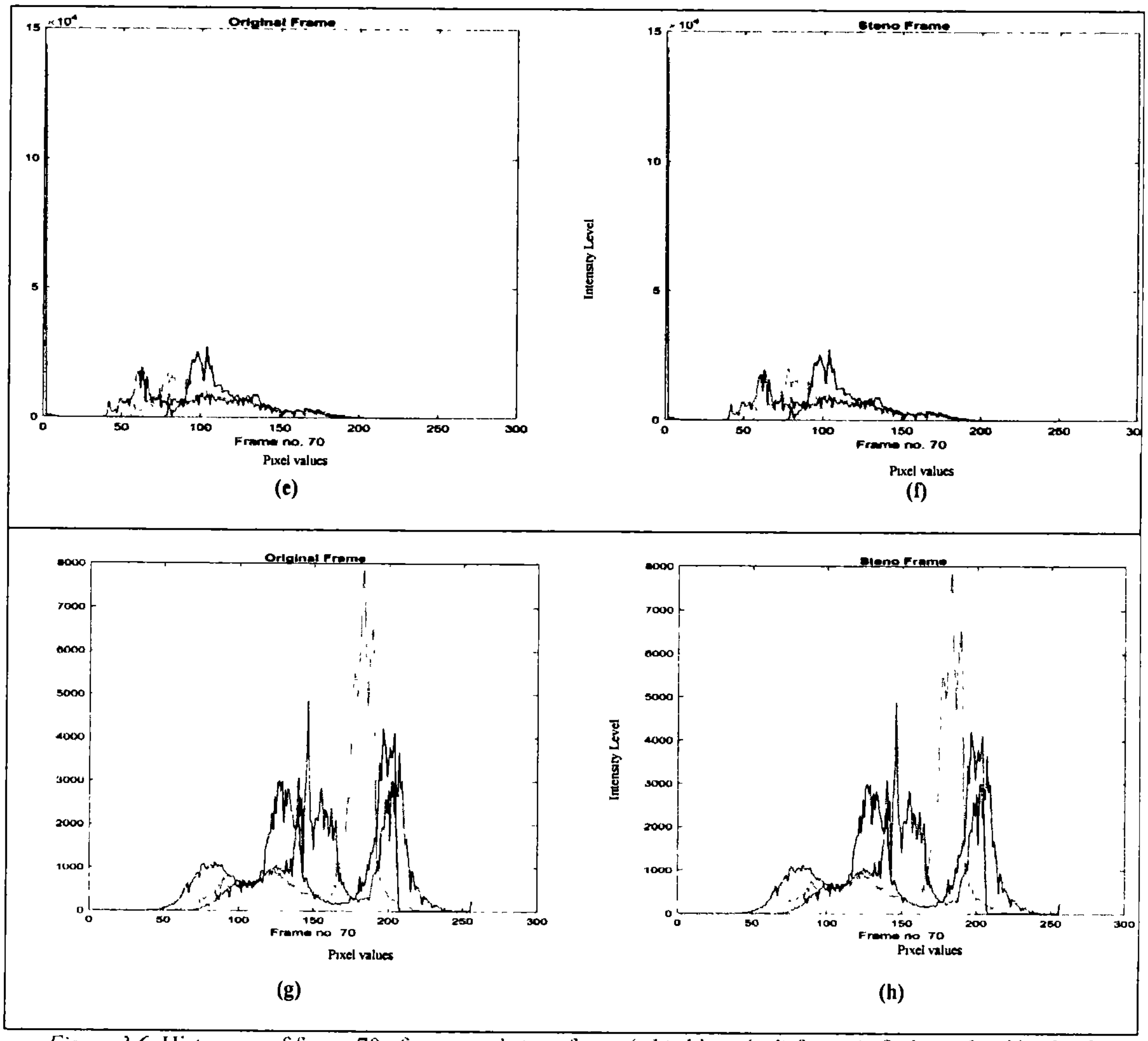
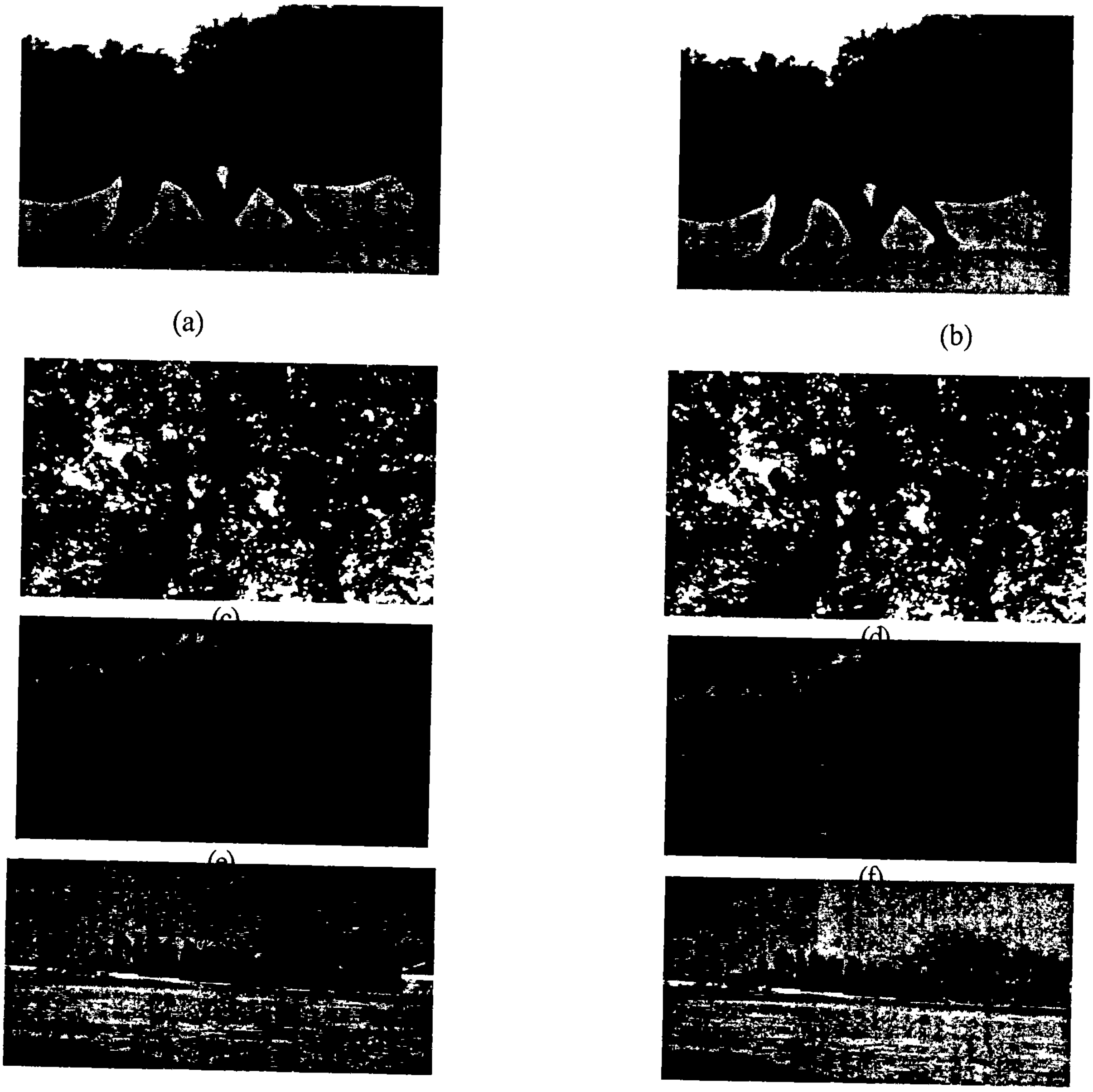


Figure 3 6· Histogram of frame 70 of cover and stego frame (a,b) rhinos (c,d) forest (e,f) sky and (g,h) school

3.5.4. Pixel correlation

Images have high data redundancy. As a result, pixels exhibit significant correlations with their neighboring pixels. These correlations should be broken by a competent picture encryption scheme. Data correlation is defined mathematically by the formula, $Corr = \Sigma_{ij} \frac{(i-\mu i)(j-j)p(i,j)}{\sigma i \sigma j}$

Here μ, is the average value i.e. the mean and σ is the standard deviation calculated over the mean for any two data sequences i and j. The correlation value of two sequences i and j is near to one if their correlations are large aside from that, it's close to zero. We examined the correlation between the cover frame and the embedded frame. The cover image has a correlation value of 1 and the stego image has a correlation value of 0. Hence, it can be stated that the suggested algorithm's encrypted image has a very low correlation.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.7 Original and stego images of frame 70 (a,b) rhino (c,d) forest (e,f) sky and (g,h) school

3.6 Security and Attacks Analysis

In the proposed solution we have embedded our encrypted data with in a DNA in selected frames of a video whose pixels were chosen by a combination of Burger Chaotic Map and linear congruential generator. The use of these techniques in video steganography can enhance the security of the hidden data in several ways. The use of DNA embedding can make it more difficult for an attacker to detect the presence of hidden data, as the data is embedded in a way that is not immediately recognizable as being different from the original video data. If the data is hidden in undetectable pixels of a video in encrypted form, it would be very difficult for an attacker to detect the presence of the hidden data without knowing the exact location of the pixels that were used to hide the data. The use of undetectable pixels means that the pixels containing the hidden data are indistinguishable from the other pixels in the video, making it much harder to detect.

Furthermore, if the data is encrypted, an attacker who is able to detect the hidden data would not be able to read it without the appropriate decryption key. This means that even if an attacker somehow manages to detect the presence of the hidden data, they would still need to break the encryption to access the data, which can be very difficult if a strong encryption algorithm and key are used.

3.6.1 How Can Burger Chaotic Map Help in Selecting Unique Pixels of a Video

The Burger Chaotic Map is a type of chaotic map that can be used for image and video encryption. In video steganography, it can be used to help select unique pixels within the video that can be used to hide data. It generates a sequence of chaotic values that can be used to select pixels within the video that are difficult to predict. These pixels can be used to hide data, making it more difficult for an attacker to detect the presence of the hidden data.

The process of using the Burger Chaotic Map to select unique pixels within a video typically involves generating the chaotic values that can be used to select unique pixels within the video. The chaotic values generated by the Burger Chaotic Map are used to select unique pixels within the video that are difficult to predict. These pixels can be used to hide the data. The data is then embedded within the selected pixels using steganography techniques. The chaotic values generated by the map are difficult to predict, making it more difficult for an attacker to detect the presence of the hidden data. Additionally, by selecting unique pixels, it is possible to increase the amount of data that can be hidden within the video without significantly affecting the video quality.

Numerous tests are performed to determine an algorithm's resistance to numerous attacks. Attacks may occur throughout the network, which must be addressed when creating a secure cryptography approach.

Plain Text Attack

This is a relatively simple attack against a cryptographic algorithm. When the sender transmits data for encryption, the attacker may intercept portions of that plaintext. Because the key is sent through a secure channel, the attacker never learns it. The attacker attempts to build the encryption technique utilized for the cipher text using some known cyphers and plaintext. This method is then used to decrypt the cipher text further. The proposed solution sends no plaintext across the network so only cipher text is sent. As key is never communicated over the network, this attack becomes very unlikely. Even if the attacker captures few frames, this attack is hard to execute since the text is encrypted and distributed in random pixels in complex frames throughout the video.

Man in the Middle Attack

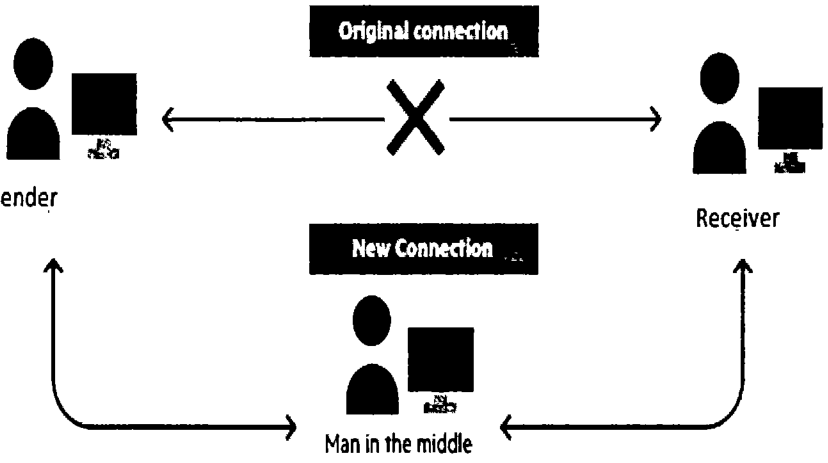


Figure 3.8. Man in the middle attack scenario.

This attack is possible if the attacker can intercept the two communicating parties' secret conversion as shown in figure 3.10. In this scenario, an attacker may intercept video transmission between the two parties, but the footage obtained will be encrypted. The encrypted data may be decoded only if both the key and the algorithm for encryption is well-known. As private key is being utilized in this scenario, it is very difficult to determine the key since the private key is never sent across a network. If the secret key is unknown, it becomes quite difficult for an unauthentic individual to retrieve the data that bolsters the suggested approach's security.

Packet Sniffing

Packet sniffing is a type of network eavesdropping where an attacker intercepts and reads data packets in a network. If the hidden data is encrypted and embedded within undetectable pixels of a video, it may be difficult for attackers to identify the packets containing the hidden data.

Traffic Analysis

Traffic analysis is the process of analyzing network traffic to determine patterns and trends. If the hidden data is encrypted and embedded within undetectable pixels of a video, it may be difficult for attackers to identify the specific packets containing the hidden data.

Brute-Force Attacks

Brute-force attacks involve trying every possible combination of characters until the correct one is found. If the hidden data is encrypted with a strong encryption algorithm, it may be very difficult for an attacker to decrypt the data using a brute-force attack.

Dictionary Attacks

A dictionary attack is a type of password cracking attack where an attacker uses a list of common words to guess a password. If the hidden data is encrypted with a strong encryption algorithm and a unique encryption key, it would be very difficult for an attacker to guess the encryption key using a dictionary attack.

3.7. Limitations

The system has been tested on videos approximately 10 seconds of duration and having a minimum resolution of 320x240 and a frame rate of 30 frames per second. Each video must consist of scenes that change. A video having frequent scene changes and above resolution will give ideal results.

3.8. Summary

The objective of this research was to introduce a more secure method of video steganography by using DNA for embedding and using a secure frame selection algorithm that would help us maintain video perceptibility. For which we proposed an improved version of DNA-based video steganography that creates an artificial DNA for embedding purpose by utilizing DNA cutting properties. Frame selection has been done introducing a new technique of scene change detection. For pixel embedding randomness has been added by using linear congruence generator, burger chaotic map and RGB channel. The proposed technique has shown better results in terms of maintaining video perceptibility that reduces the chances of suspicion. Further, a high embedding efficiency, reduced MSE and reduced PSNR show the success of this technique in comparison with the base technique. This paper proves that with an intelligent frame selection and improved embedding techniques we can transmit our steganography videos securely over a communication medium.

Chapter 4

Detection of Upscale-Crop In Steganographic Video

4. Introduction

This chapter discusses our third problem and provides its solution. It starts with the introduction on video forgery and its types. After discussing the basics of video forgery, it presents the proposed solution for the detection of resampling using pixel-correlations and inconsistencies in background noise.

4.1 Introduction to Video Forgery

Every day, social networking sites receive millions of photographs and videos. As a result, our reliance on statistics to portray reality has increased. Photographs and films are also used as evidence in intelligence, journalism, investigations into insurance claims, and legal proceedings. However, with recent advancements in image and video editing software such as Lightworks, WeVideo and many others the authenticity of these images and videos must be called into question. These forged images and videos leave no visual clues, leading to the misdirection of a large number of people, as forged images and videos spread widely through social networking sites on a daily basis. These soft wares, along with several other forgery methods, have made it possible for even inexperienced users to quickly remove an object and replace it with an object from another video source, or insert an object made by a graphics designer. The presence of such high-accuracy forgery tools and techniques, as well as their ever-increasing demand and supply, has posed a continuous challenge to the field of image and video tampering detection. Fei et al. [30] implemented a cryptographic hash to prevent frame insertion and manipulation. S. Chen et al. [31] used chaotic systems and DCT to verify surveillance footage tampering. Timing information in frames is interpreted as chaotic system characteristics, resulting in a noise-like watermark in the block-based discrete cosine transform domain. Demodulating stored data with a maximum likelihood estimator. Temporal manipulation is detected by comparing recovered and experimental time data. The extracted watermark reflects spatial changes in the original document. D. Xu et al. [32] split a watermark image into binary images and put them into video bit streams. This strategy defeated frame averaging, scaling, filtering, frame dropping, and rotation.

Qianwen et al. [33] established an automatic jump-cut detection method for analysing video modification and tampering. HVS inspired 4-EGSSIM, which enhances gradient images with logarithmic transformation. MSE and peak SNR are utilised to evaluate results (PSNR). G.H. Chen et al. [34] propose Gradient Structural Similarity index to replace SSS (SSIM). It inputs picture gradients. Nercessian et al. [35] created the 4-SSIM and 4-GSSIM for smooth areas, edge change, texture, and edge preservation. A. Gironi et al. [36] used algorithms to determine frame deletions and additions. As frames were added or withdrawn, they re-encoded a static group of pictures (GOP). Frame insertion or deletion is unknown.

P. Bestagini et al. [37] detect substituted areas, which are replaced with fixed pictures or the same video at different times. This approach fails while switching a video sequence. Singh et al. [38] use absolute and relative local correlations. Video frames were watermarked using local relative correlations. C.-Y. Liang et al. [39] suggested geographical and temporal domains influence shot segmentation. R.D. Singh et al. [40] detect interframe forgeries using pixel correlation and noise pattern analysis. M. Fallahpour et al. Misindexed macroblocks help detect harmless recompressions, distortions, and filters. Malicious intent is discovered if macro blocks appear frequently in the same frame and place. Hyun et al. [42] use Sensory Pattern Noise (SPN) and the MACE filter to detect video frauds. Scalar factor and correlation coefficient find various forgeries, including partial modification, video alteration, and upscale-crop. Kobayashi et al. [43] identified faked video by noise discrepancies. A linear Noise Level Function (NLF) is used to assess the connection between the collected sounds to detect frauds. Fayyaz et al. [44] devised a method for detecting temporal copy-paste in paintings. Adaptive DCT filtering compares each video frame's noise residue patterns to the SPN to detect forgeries.

All these strategies have been applied on the detection of any kind of tampering in videos whereas no consideration has been given to the detection of any kind of tampering in steganographic videos. On of the reasons for this lack of research is that video steganography is a new field compared to image steganography.

Without secure steganography techniques, sensitive data contained in videos can be readily tampered with or removed. For a stego video, video tampering, which includes the addition of new frames, the deletion of existing frames, or the change of critical data, has garnered minimal attention. In the case of a stego video, interfering with the cypher text may corrupt it, resulting in the loss of critical information. The process of identifying these frauds in a cypher text video continues.

4.3.2 Proposed Approach

There has been a lack of research in the domain of detecting video resampling in a steganographic video. In this chapter, we will look at how upscale crop and splicing kind of tampering is detected in steganographic videos by utilizing the techniques mentioned for detecting these forgeries in a non-steganography video. Our focus will be on the approach described in [44], in which resampling is detected using Pixel-Correlations and inconsistencies in background noise caused by the recording medium. They have examined the correlation between pixels and the noise inconsistencies present with in a video to detect intra-frame forgeries of the type known as splicing and upscale-crop which crops outer parts of the frames. For this the resampling detector used are the Modified-Gallagher (MG) Detector and another detector is the F-MG Detector (Fractional MG) [51]. The same authors have also provided this strategy of detecting and localizing copy-paste forgeries in digital video by utilizing the Sensor Pattern Noise Correlation (SPNC) in [52].For this we have used we have used the dataset by Qadir et al. [60] developed for the Surrey University Library for Forensic Analysis as a video dataset for testing video forgery detection techniques (SULFA). It is entirely comprised of forgery-based videos of the copy-move variety. It's an online dataset of 150 movies gathered from stationary cameras and is available for free use. An individual video is 10 seconds long, has 30 frames per second, and a 320x240 pixel resolution.

Pixel-Correlations: Resampling alters the pixel-level relationships in digital content, therefore, upscale-crop forgery and the splicing forgeries can be detected by evaluating the correlations between pixels.

Noise Variations and Inconsistencies: Random noise insertion in tampered content is one approach for concealing evidence of tampering, as noise obstructs the detection of resampling by conflicting with exact forensic evidence sought by the resampling detectors. So, adding any random noise locally to the tampered frames or its regions, the upscale-crop and splicing traces can be removed. The observational evidence provided by these two artefacts will be evaluated in order to achieve a final determination on the legitimacy of the material. The complete system organization is depicted in Figure 4.1.

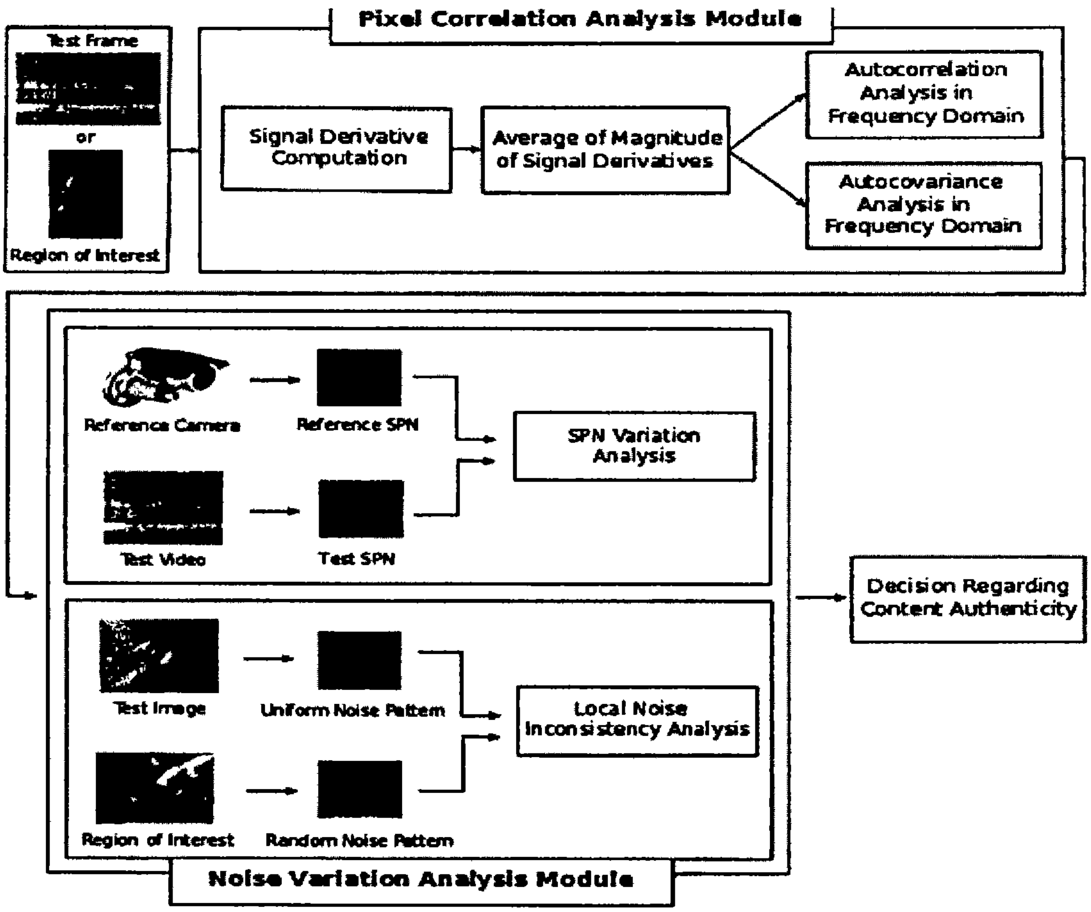

Pixel Correlation Analysis Module
Test Frame
or
Region of Interest
Signal Derivative Computation
Average of Magnitude of Signal Derivatives
Autocorrelation Analysis in Frequency Domain
Autocovariance Analysis in Frequency Domain
Reference Camera
Reference SPN
SPN Variation Analysis
Test Video
Test SPN
Decision Regarding Content Authenticity
Test Image
Uniform Noise Pattern
Local Noise Inconsistency Analysis
Region of Interest
Random Noise Pattern
Noise Variation Analysis Module

Figure 4 1: The splicing, upscale crop forgery detection system

4.3.2.1 Resampling Detection:

Not only is resampling the major technique employed in upscale crop forgeries, but it also leaves evidence of post-production modifications. Up sampling is essential in the case of upscale-crop to extend the cropped frames and ensure frame resolution uniformity between the forged and unforged frames in the video. This introduces specific relationships or correlations between adjacent pixel groups. By detecting these unexpected pixel correlations, we may determine the presence of resampling fingerprints in the frames of a given film. The procedures needed in detecting resampling are illustrated in Fig.4.2 below.

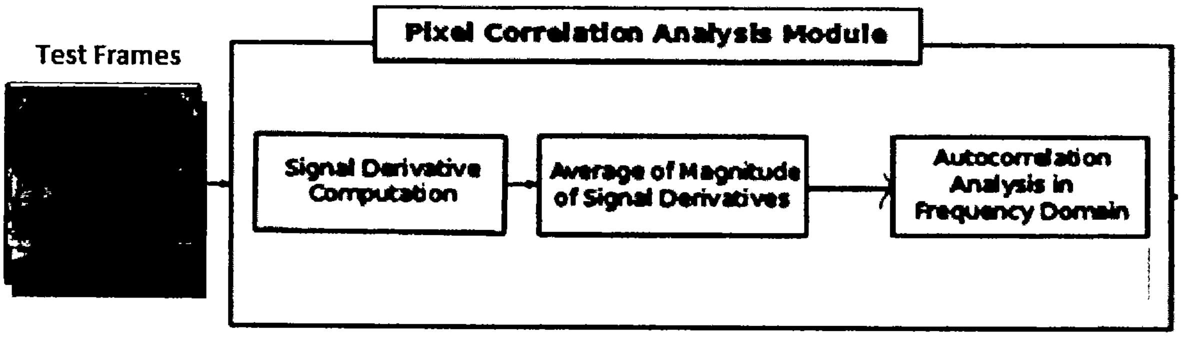

Test Frames
Pixel Correlation Analysis Module
Signal Derivative Computation
Average of Magnitude of Signal Derivatives
Autocorrelation Analysis in Frequency Domain

Figure 4.2. The resampling detection system

4.3.2.2 Examining Pixel Correlation using Peak Analysis:

Frequency domain peak analysis is a widely used forensic technique because it enables us to visualize the statistical features of pixels as clearly evident artefacts in the frequency domain. When checking for splicing, the suggested technique computes the nth order derivative of each row in the entire frame. In case of a specified Region of Interest (ROI) or frame sub-block, it uses the same approach i.e. computes the nth order derivative. The pseudo-variance signal is then calculated, which is the column wise mean of these derivatives' magnitudes, rather than the row wise mean as in [40]. Correlation and covariance analysis is the final phase, which is performed on all signals of variance' produced in step 2. The analysis is carried out in the DCT and Fast Fourier Transform domains (FFT). In algorithms step 2 and step 3 the absolute (magnitude) and real values of signal derivatives and frequency coefficients are determined. The MG Detector's algorithm is as follows: -

```
1. for every frame $f(x, y)$
       if checking for upscale-crop forgery
           $D_r^n\{f(x,y)\} = \partial nf(x, y) / \partial xn\partial yn$ /*nth order derivatives for entire frame in x (row)-direction */
       else for region-level splicing detection
           $D_k^n\{f(x,y)\} = \partial nf\,k(x, y) / \partial xn\partial yn$ /* nth order derivatives for ROI or $k^{th}$ frame sub-block*/
   end
2. for each column c of $n^{th}$ order derivative matrices
           $V(x) = \Sigma|D^n\{f(x,y)\}|$ /* ID signal of variances */
   end
3. for every $V(x)$
           $CVv(k) = \Sigma\,(V(i + k) - \bar{V})(V(i) - \bar{V})$ /* auto covariance sequence */
           $CR_v(k) = CVv(k) / CVv(0)$ /* autocorrelation sequence */
       $FS_{MG} = DCT(CRv(k))$ /*DCT coefficients of autocorrelation sequence */
           $FS_{MG(F)} = FFT(CRv(k))$ /* FFT coefficients of autocorrelation sequence */
   end
```

The above algorithm gives the $FS_{MG}/FS_{MG(F)}$values.

1. Time Complexity

To analyze the time complexity of this algorithm, we need to consider the time required for each step and the number of iterations or computations performed within each step. This includes the calculation of derivatives, summations, variances, covariance, and the DCT or FFT operations. Additionally, the algorithm may also involve memory usage and space complexity, which depends on the size of the input data and the storage required for intermediate results, the number of operations performed in each step.
If we divide the algorithm into its individual steps for finding the complexity then the complexity of each step depends on the size of the frame and the chosen order of derivatives (n). The computation of derivatives and variance involves iterating through each pixel or column, resulting in a complexity of O(N), where N is the total number of pixels or columns in the frame. The computation of the auto-covariance and auto-correlation sequences also involves iterations, resulting in a complexity of O(K), where K is the length of the sequences. The complexity of the DCT or FFT computation depends on the specific implementation, but it is generally O(K log K) or better. Overall, the complexity of the algorithm can be represented as O(N) or O(K) depending on the dominant step which means that the overall time complexity of the algorithm can be approximated as O(N) or O(K) based on which step in the algorithm has the highest computational complexity.

In algorithm analysis, we often express the complexity using the Big O notation, which gives an upper bound estimation of how the algorithm's performance scales with the input size. The notation O(N) represents linear complexity, where the execution time increases linearly with the size of the input (N), while O(K) represents complexity that depends on the length (K) of a specific sequence or set of operations.

So, depending on whether the dominant step in the algorithm is related to the number of pixels/elements (N) or the length of sequences (K), we can approximate the overall complexity of the algorithm accordingly. It is important to note that this approximation assumes that the other steps have lower complexities or can be considered constant-time operations in comparison i.e.in the context of the given algorithm, the approximation of O(N) or O(K) for the overall complexity assumes that the other steps in the algorithm have lower computational complexities or can be considered as constant-time operations.

In algorithm analysis, we often focus on identifying the most time-consuming or dominant step that significantly affects the overall complexity. The approximation of O(N) or O(K) is based on this dominant step. However, it is important to recognize that there may be other steps in the algorithm that contribute to the total execution time, but their complexities are relatively smaller or constant.

By assuming lower complexities or constant-time operations for these other steps, we simplify the analysis and focus on the primary factor influencing the overall complexity. This assumption helps provide a reasonable estimation of the algorithm's performance and facilitates understanding of its scalability with varying input sizes i.e. that in the context of the given algorithm, the approximation of O(N) or O(K) for the overall complexity assumes that the other steps in the algorithm have lower computational complexities or can be considered as constant-time operations.

2. Running Cost

The running cost of an algorithm is typically measured in terms of time complexity. To evaluate the running cost of this algorithm, we need its time complexity. Once we have the time complexity of each step, we can estimate the overall running time based on the input size and the efficiency of the hardware. Time complexity represents the amount of time taken by an algorithm to run as a function of the input size. It is usually denoted using big O notation (O()).

Our algorithm has a time complexity of O(N), where N represents the input size, it means that the running time of the algorithm grows linearly with the size of the input. In the case of this algorithm, the size of the input is determined by the dimensions of the frame, which in one of the examples is given as x = 480 and y = 380. Since the algorithm's time complexity is O(N), the running time would be proportional to the size of the input, i.e., O(N) or O(480 * 380).

Given the hardware specifications of my machine (Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz, 16 GB RAM), it handles the algorithm efficiently.

3. FSMG/FSMG(F) Plots

The obtained FSMG/FSMG(F) are plotted after step 3. Due to the lower correlation between the pixel intensity values in normal frames and those in up-sampled frames, autocorrelation sequences for up-sampled frames generally contain a large number of peak values whereas for frames with high autocorrelations these are near to zero when plotted. Figure 4.3 shows the complete frequency spectrum of the FSMG values obtained from real and resampled video frames, respectively.

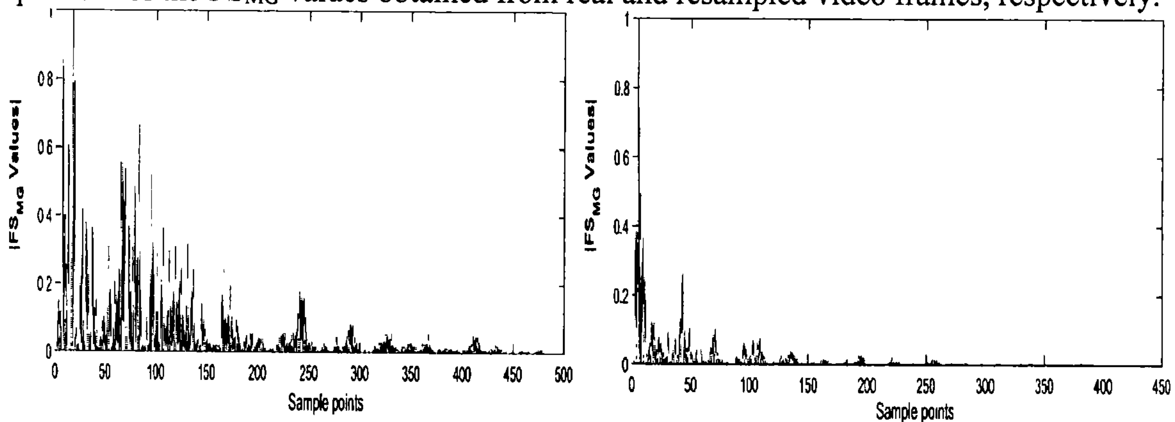


Figure 4.3: (a) Frequency spectrum of original frames autocorrelation sequence (b) Bilinearly up-sampled frames autocorrelation sequence.

The changes in the patterns of normal and tampered movies can be seen without resorting to complex peak analysis techniques. Except for a few high-amplitude peaks in the spectrum of resampled video frames above in fig 4.3, all other peaks have values extremely near to zero. The detector gives finest results for derivatives of second, third, and fourth order. When n > 4, the distinction between the original and resampled content's frequency patterns begins to blur in some circumstances.

Region/Section Level Splicing:

Any frame can be divided into non-overlapping sections or blocks and the method applied individually to each sub-block. For instance, in Fig. 4.4(a), two ROIs i.e. region of interest have been selected, denoted by the two rectangles, to identify evidence of splicing. The right ROI is a 45 x 65 pixel rectangle. The ROI in the same picture below on the left is a region 65 × 45 pixel containing forged content spliced into the image using bicubic interpolation with scaling factor SF = 1.4.

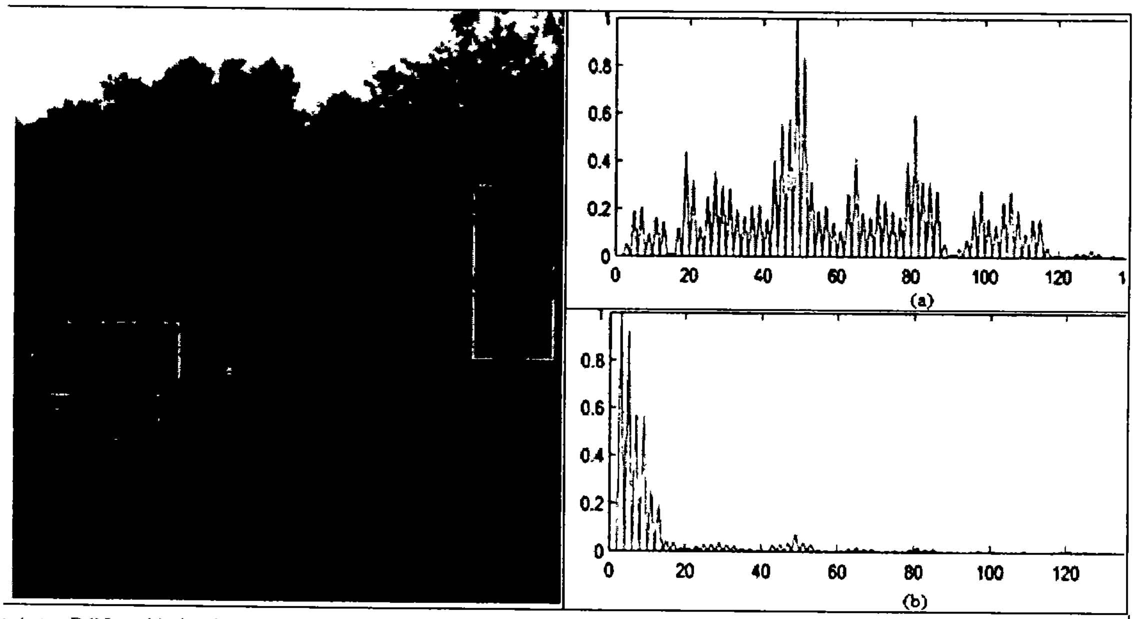
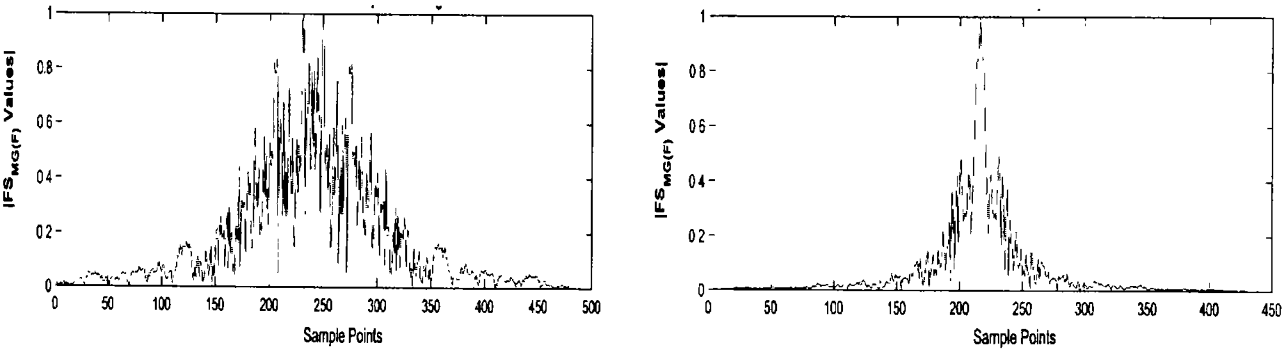


Figure 4.4. (a) ROIs , (b) the frequency spectrum of the right ROI (original content), and (c) the frequency spectrum of the left ROI (spliced content).

The plots illustrate that the original and spliced material display totally separate patterns, which enables them to be distinguished simply and accurately. As seen in Fig. 4.5(a) and (b), resampling produces apparent variations in the frequency spectra of genuine and resampled movies.



(a) (b)

Figure 4.5: FFT coefficients plots for Fig 4.13 (a, b) autocorrelation sequences for (a) riginal and (b) resampled frames

4.3.2.3 Variations in the Sensor Pattern Noise and Analysis of Inconsistency in Local Noise

Our resampling detection technique detects noise as the second artefact. Each digital camera records a unique type and amount of noise in each frame that is generated by the imaging devices known as the Charge Coupled Device acronym CCD. These devices are composed of a many photon detectors containing several silicon wafers. Due to the fact that silicon wafers are not identical, their light sensitivity vary. As a result, each photon detector produces a different noise called Sensor Pattern Noise i.e. the SPN, which appears in every frame it captures.

While the noise among different capturing devices varies naturally, it also exhibits a unanimous configuration over a sequence of successive frames captured by the same device. Any variation in this pattern is the first indicator of an unapproved alteration. A reference SPN can be retrieved from the recording device or it can be extracted from any clip shot by the same camera. To hide the impact of resampling if noise is added to few of frames or the even to the whole video, the difference in the noise patterns of these frames will clearly show the presence of fraudulent activity. Adding random noise even to a tiny part of the frame to obscure the evidence of region-level splicing, will disrupt frame's identical noise pattern. As a result, variations in local noise suggest the presence of a fabricated region in such frames. The autocorrelation of the reference and test SPNs, as well as their variations, aid in distinguishing genuine videos from those with random SPN fluctuations and local noise discrepancies.

Sensor Pattern Noise SPN Analysis for Upscale-Crop Detection

SPN can be calculated for a video sequence using [50]. I_1, I_2, I_3... I_n, represent the sequence of N frames in a video then SPN is calculated as in equation 4.1 below

$$\text{SPN} = \sum_{k=1}^{N} \frac{W_k \hat{I}_k}{\left(\hat{I}_k\right)^2} \qquad 4.1$$

where $\hat{I}_k = F(I_k), \quad W_k = I_K - \hat{I}_k.$

Here, F denotes the wavelet-denoising filter [52]. The reference and test SPNs are calculated using the aforementioned equation, to compute autocorrelation sequences. S symbolizes sensor pattern noise, we may construct auto covariance and autocorrelation sequences using below equation 4.3

$$CV_S(k) = \sum_i \left(S(i+k) - \bar{S}\right)\left(S(i) - \bar{S}\right) \qquad 4.2$$

$$CR_S(k) = \frac{CV_S(k)}{(CV_S(0))} \qquad 4.3$$

4.3.2.4 Plots Evaluation:

If the noise patterns in distinct sections of a frame change from those in preceding or subsequent frames, the plots of temporal correlation values will reveal these differences. Correlation patterns of test and reference SPN extracted from the suspicious videos are shown in Fig. 4.6 The differences in the plots enable identification of the suspected tampered video frames.

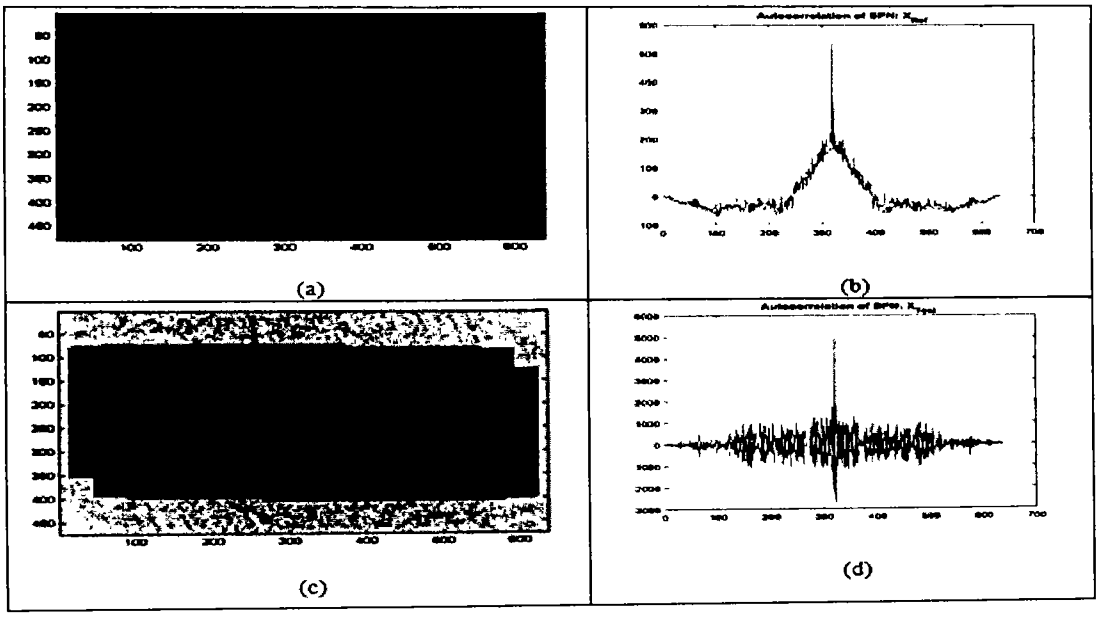


Figure 5 (a) Autocorrelation sequences for the reference XRef (c) Autocorrelation sequences for the test SPNs XTest. (b) and (d) are plots of XRef and XTest, respectively.

4.3.2.5 Dataset:

The experiment used videos as the dataset given in [50]. The Surrey University Library for Forensic Analysis (SULFA) was unveiled in IPR 2012 and is freely accessible for academic pursuits. The Library is bifurcated into two segments i.e. the original videos that aid in testing algorithms for camera identification and device linkage and the other comprises of the forged videos that facilitate the examination of forgery detection algorithms. SULFA, a unique resource in the field, has been structured specifically to benchmark video forensic techniques, focusing primarily on camera identification and integrity verification.

Accessible via the University of Surrey's website, SULFA hosts an array of both original and manipulated videos. The collection includes around 150 videos, sourced from three different cameras: Canon SX220 (codec H.264), Nikon S3000 (codec MJPEG), and Fujifilm S2800HD (codec MJPEG). Every video, approximately 10 seconds in duration with a resolution of 320x240 and a frame rate of 30 frames per second, was filmed considering both temporal and spatial video aspects.

Various intricate and simplistic scenarios have been captured both with and without the use of camera supports like tripods to mirror real-life situations. The library also includes nine original videos from each camera source, which have undergone tests with Photo Response Non Uniformity (PRNU) based camera identification methods. Currently, SULFA encompasses videos featuring cloning or copy-paste forgery. Comprehensive information regarding the manipulated regions is provided for each of these forged videos.

4.3.2.6 Results Discussion:

As the module identifies upscale crop detection in stego videos, it was also tested on stego videos from module 1's output i.e. chapter 3, which makes a stego video. MATLAB R2012a was used to conduct all experiments. The test data for upscale-crop detection was created in two distinct ways by cropping and resampling video frames. 50% of the test frames were cropped by eliminating material from their boundaries and then expanded using a scaling factor of 0.5 to 0.9. Further, the system is limited to detecting tampering of type upscale crop on frame level and region level only.

a. MG detectors Qualitative Findings:

Figure 4.7 correspond to the frequency spectrum of the 30 X 30 pixel ROIs depicted in Figure 4.13 (a)

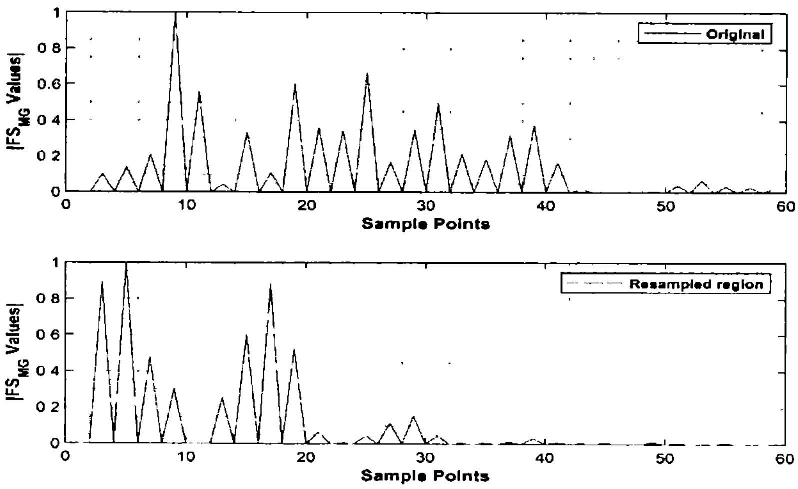


Figure 4.7 Figure shows the frequency spectrum of auto covariance sequences for the original region (a) and the resampled region (b) for ROIs of 30 x 30 pixels.

b. Plots of Correlation and Covariance for Similarity Examination.

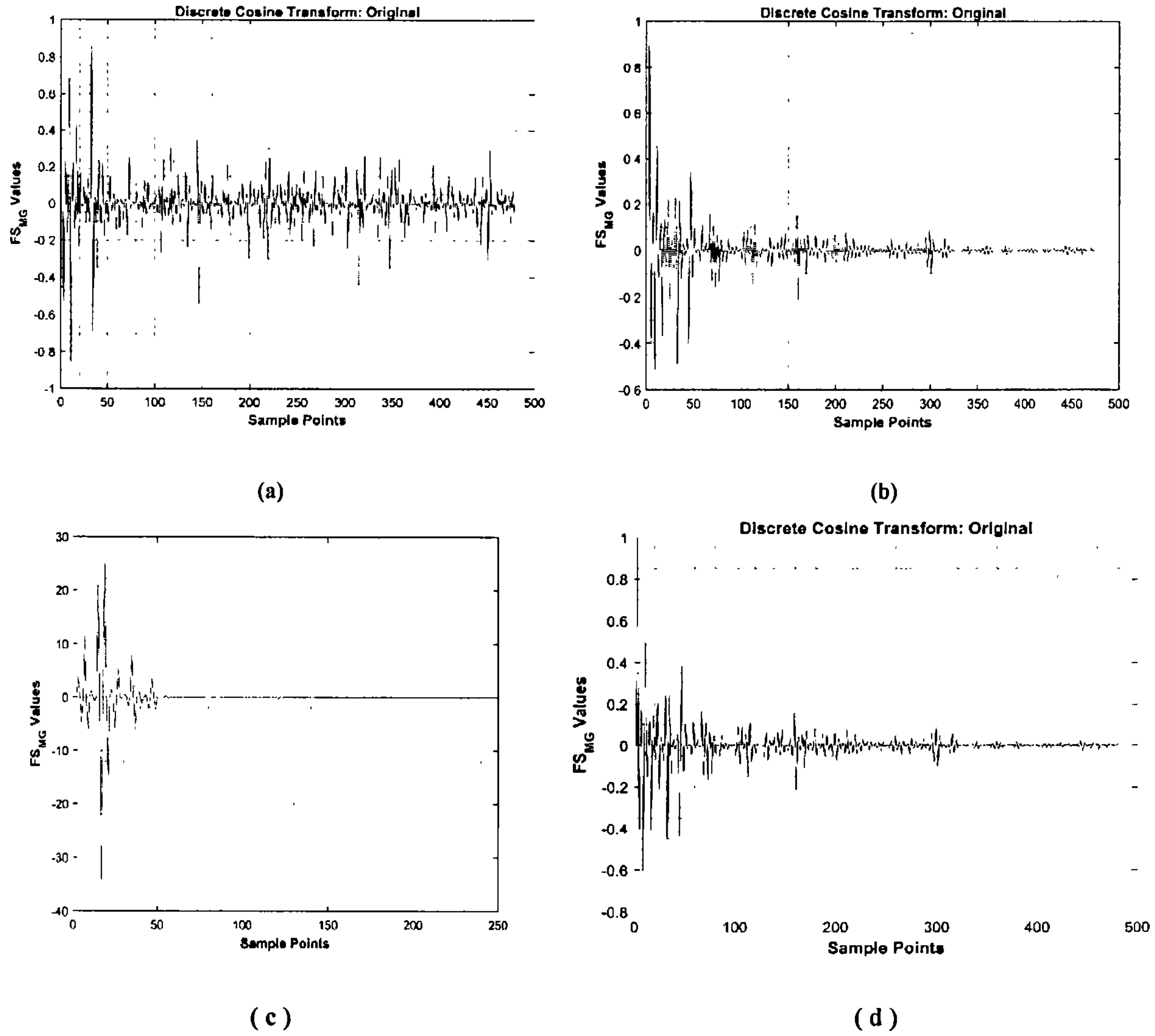
Correlation analysis is considerably superior to covariance analysis for detecting forgeries, as indicated previously in the sub-section proposed technique for the MG detector

Resampling generates statistical inconsistencies between the correlation and covariance of forged videos and this phenomena can be utilized as an additional criterion for spotting tampered frames of a steganography video.

c. System Performance with Existing Noise

Noise of different known types were added to evaluate the systems performance in the presence of noise. For this Poisson noise with $\sigma^2 =$ 0.02, 0.04, 0.06, Gaussian and speckle noise with $\sigma^2 =$ 0.02, 0.04, 0.06 and Salt and pepper noise with different densities d =. 0.01 and 0.02 were added to the original and resampled test frames to assess the system's performance in the presence of noise.

Fig. 4.8 illustrates that detector's capacity to recognize resampled frames is not affected by the addition of random types of noise but it does effect the frequency spectra of original frames.

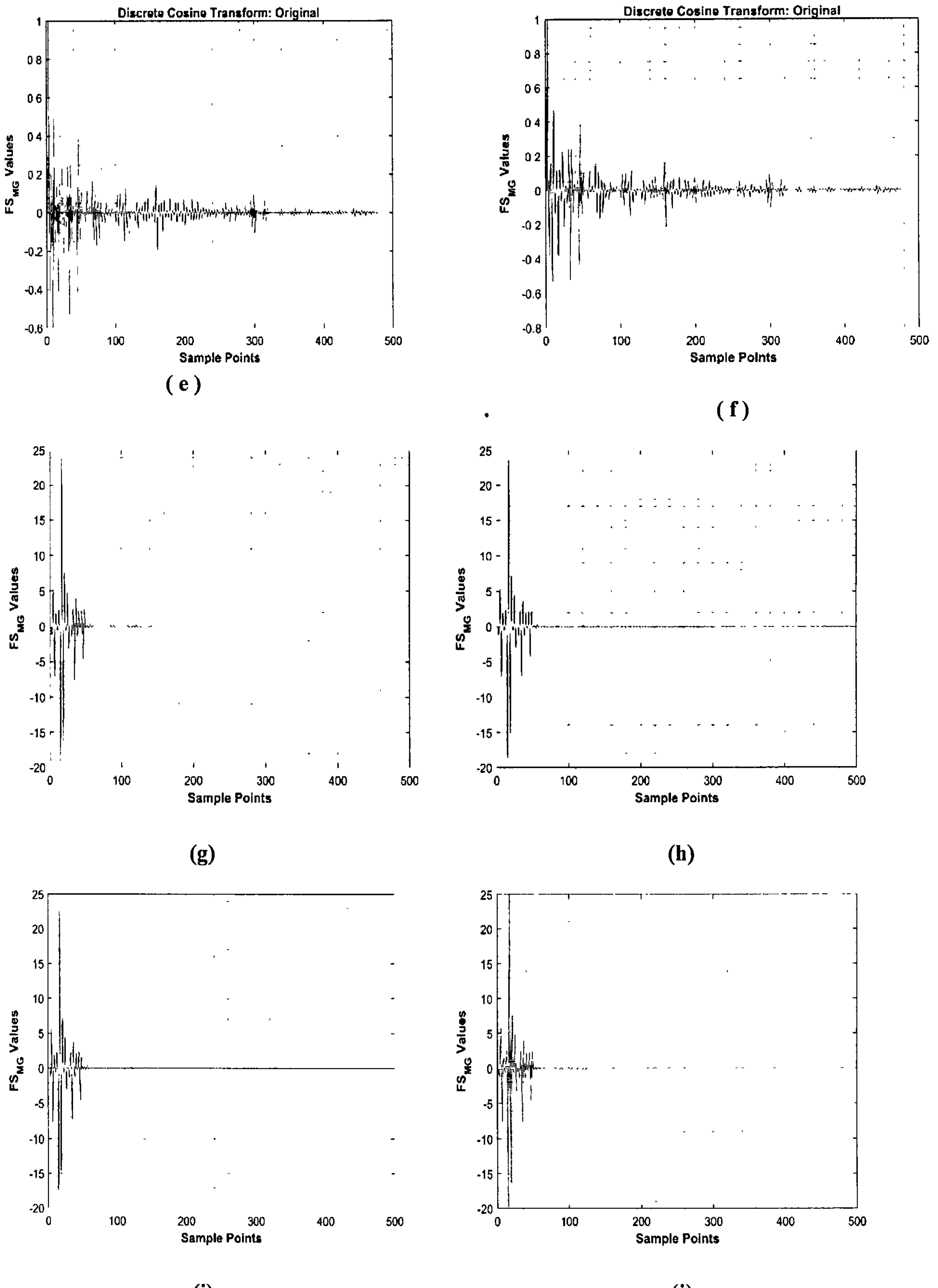


(a)

(b)

(c)

(d)


Discrete Cosine Transform: Original
FSMG Values
Sample Points
Discrete Cosine Transform: Original
FSMG Values
Sample Points
FSMG Values
Sample Points
FSMG Values
Sample Points
FSMG Values
Sample Points
FSMG Values
Sample Points

(e)

(f)

(g)

(h)

(i)

(j)

Figure 4.8. (a) Illustrates the frequency spectrums of the original and the enlarged frames of Fig 4.13 respectively. (c), (d),(e) and (f)show the original frames frequency spectrum in the presence of the Gaussian noise with the value σ2=0.02. Poisson noise, salt and pepper noise σ2=0.05 and speckle noise σ2=0.04. (g).(h)(i) and (j)show the frequency for the resampled frames respectively

Both Gaussian and salt & pepper noise cause the original frame's frequency spectrum to follow a similar pattern to that of a resampled frame, which might result in an original frame being identified as a false positive. Denoising was accomplished by adjusting the filter's sigma value in stages of ten from 20 to 60. Fig. 4.9 shows spectrums of original frames with Gaussian and speckle noise, as well as the denoised version of these frames. These figures indicate that after denoising, the original frames' frequency spectrums return to original patterns and can be differentiated from resampled frames.

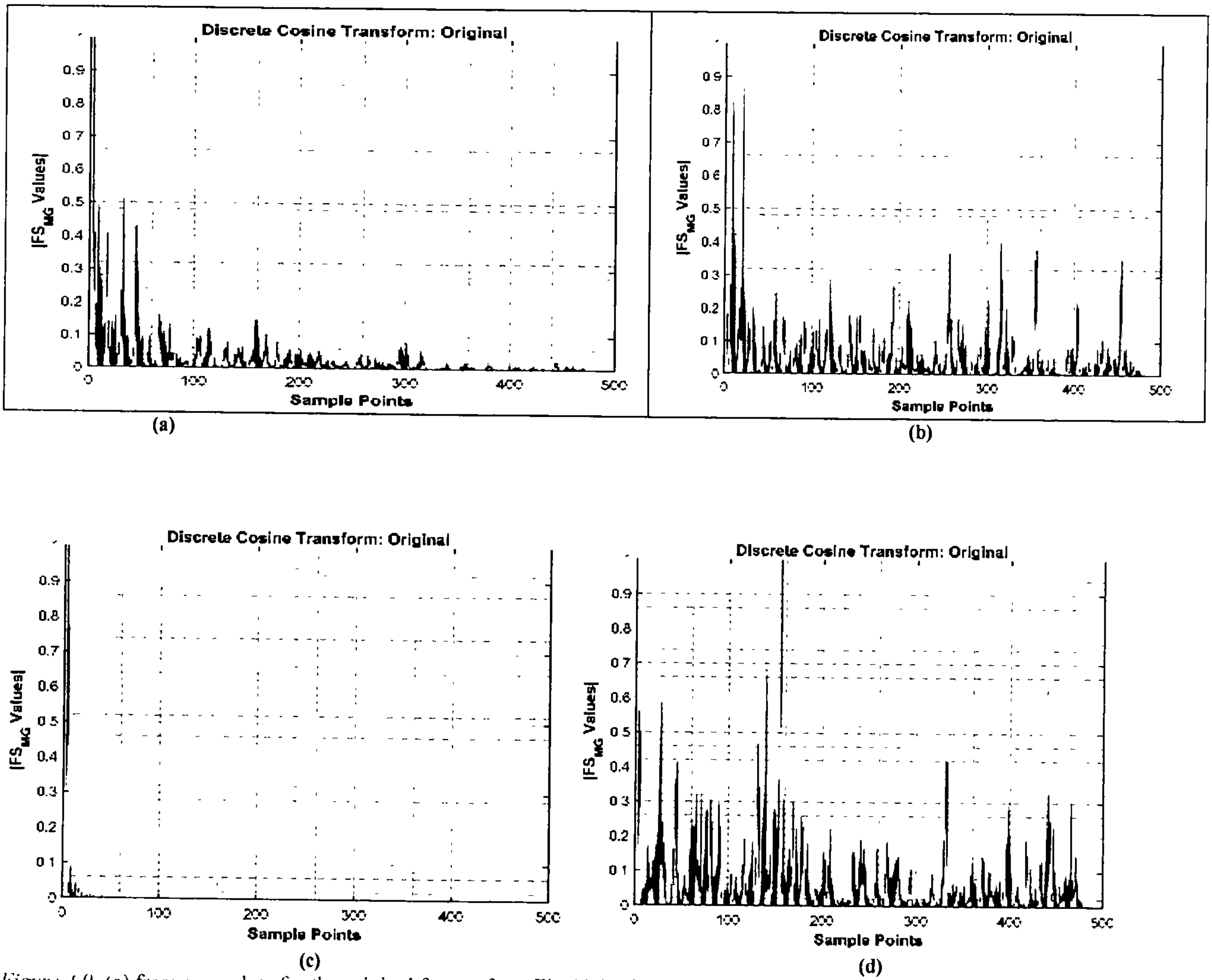


Figure 4.9. (a) frequency plots for the original frames from Fig 41,(c) frequency plots for the original frames from Fig 41both (a) and (c) with Gaussian noise σ2=0.02and speckle noise σ2=0.06.(b) and (d) frequency plots for the denoised versions of these frames for denoising sigma 50 has been used.

Independent of the characteristics of the underlying content, the characteristic frequency spectra of resampled videos are the most significant signs of manipulation. They provide substantially more dependable evidence for detecting forgeries accurately. As a function of different scaling factors, Table 4.1 plots the average detection accuracies of the recommended detectors, as well as the accuracies obtained using the detectors presented in [44], [45], and [46]. The results demonstrate that the proposed detectors are capable of detecting interpolation over a wide variety of scaling factors while preserving their efficacy when data is compressed with very low quality factors.

Algorithm	**QF**	**Scaling Factor**							**Average**
		1.05	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**Accuracy**
MG Detector [50]	100	100	100	100	100	100	100	100	100
	80	98.7	99.3	99.3	100	100	100	100	99.61
F-MG Detector[50]	100	100	100	100	100	100	100	100	100
	80	99	98.4	98.3	99.5	100	100	100	99.31
Gallagher [51]	100	25.7	27.7	28.2	29.2	30.7	35	40.7	31.03
	80	19	19	22	21.5	22.8	28.2	31.9	23.5
Kirchner and Gloe[52]	100	70.4	71.8	76.2	77	80.8	81.5	89.6	78.21
	80	62	63	68.8	72.5	75.8	75.2	83	71.47
MG Detector	100	100	100	100	100	100	100	100	100
	80	98.3	99.1	99.7	100	100	100	100	99.56

Table 4.1 Demonstrates the detection accuracy of the proposed detector in comparison with [50], Gallagher [51], Kirchner and Gloe[52]for a variety of quality and scaling factors

Table 4.1 demonstrates the detection accuracy of the proposed detector in comparison with [50], Gallagher [51], Kirchner and Gloe[52]for a variety of quality and scaling factors.

4.4 Summary

The suggested approach provides a reliable validation of steganography videos by detecting resampling patterns using traditional peak analysis procedures. The technique can detect forgeries in small regions in a frame as well as in the entire frame. The proposed detectors can reliably identify all forgeries with higher than 99 percent accuracy, for quality factors between 80 and 100. The technique gives higher than 98 percent detection accuracy, even low QFs of 60. Additionally, the proposed technology mitigates noise's effects. The type and quantity of deteriorating noise contained in supplied content has no bearing on the system's forensic capabilities.

Chapter 5
Conclusions and Future Work

5 Conclusion

The purpose of this research was to introduce an improved algorithm for creating a steganography video. The newly devised algorithm should have the capability of storing data in such a way that it is not detected by the naked eye nor can be cracked by any eavesdropper. Further, it was proposed to find a mechanism to detect upscale crop and resampling in a steganographic video for the purpose of identifying a tampered steganographic video.

We have proposed a new algorithm for storing text in a steganographic video by embedding our data in DNA. For this the data was first encrypted with an asymmetric cipher, converted to binary codes and then embedded in DNA using our devised algorithm. Once our encrypted data is ready then it will be embedded in the video. For video embedding, we have also devised a new algorithm. Embedding is not done on any random location. Rather, we have choose frames that involve include scene change information so that our hidden data can be misunderstood as a scene change information. For such frame selection we have devised a new algorithm that will select only those frames of video that involve scene change. These peculiar frames are separated from the video and set aside for embedding. To make the embedding secure we have selected not al pixels of the video but only few random pixels by using burger chaotic map and linear congruential generator. The combination of these two algorithms adds another layer of security to our solution by providing us random locations in a frame that is already having too much information to comprehend. Data is not embedded in all locations, nor in all frames. It is divided among few selected pixels of few selected frames out of the pool of frames containing the complex frames. This adds another layer of security as all the frames are not utilized. The data is also embedded at ideal locations within the few RGB components of the selected pixels. After embedding, we generate our stego video by combining the stego and non stego frames. The resultant video has shown better performance in term of visual perceptibility, reduced MSE and higher PSNR show the success of this technique in comparison with the base technique.

Our third contribution is utilizing the technique proposed in [44] that involves the detection of tampered frames in a stego video, this detection was never done in a stego video before. Once a stego video is ready and during transmission it is tampered such that any of its frame is resampled or cropped, it can be detected by analyzing the unusual pixel correlations with in frames. For this pixel correlations were analyzed by calculating the DCT and FFT coefficients of auto correlation variances known as the $FS_{MG}/FS_{MG\ (F)}$. The subsequent frame spectrum look significantly dissimilar when its FS_{MG} values are plotted for frames of the original and resampled videos, in both cases. The majority of the peaks have extremely small magnitudes, with the exception of a few extremely high amplitude peaks in the resampled frame spectrum. By assessing pixel discrepancies and SPN variants, the proposed approach assists us in precise validation of resampled videos by identifying resampling patterns using conventional peak analysis procedures.

5.1 Future Work

There is no end to research. There are several potential areas for future research in this thesis. Firstly, one area of future research could focus on enhancing the security of the steganographic technique. The thesis chapter 3 mentions that if the DNA used for frame selection is compromised, the hidden and unencrypted data will be exposed. Therefore, it would be valuable to explore methods to strengthen the security of the DNA-based steganography, such as using advanced encryption algorithms with reduced time complexity or incorporating additional layers of security measures.

Secondly, the imperceptibility of the stego video is an important aspect to consider. The thesis chapter 3 mentions the use of an intelligent frame selection algorithm to improve video imperceptibility. Further research could investigate different techniques to optimize the imperceptibility of the stego video, such as exploring advanced image processing algorithms or developing new algorithms specifically tailored for video steganography.

Future research could explore the effectiveness and efficiency of different combinations of cover media, such as exploring the use of other biological materials or incorporating other types of complex frames (e.g., frames with complex patterns or textures). This could help to further enhance the security and imperceptibility of the steganographic technique.

Furthermore, the paper highlights the importance of selecting a proper cover media in video steganography. Future research could focus on developing methods or algorithms to automatically select the most suitable cover media for a given steganographic task. This could involve considering factors such as the characteristics of the cover media, the desired level of imperceptibility, and the security requirements.

Future research could build upon this work by proposing and evaluating new algorithms or techniques for dual cover steganography. This could involve exploring different types of cover media, refining the embedding and extraction processes, or investigating the impact of different parameters on the performance of the steganographic technique. Other forms of messages such as audio, visual and video can also be tested as the medium that has to be embedded. Further research could focus on enhancing the security, imperceptibility, and efficiency of the steganographic technique. This could involve exploring advanced encryption algorithms, optimizing image processing techniques, investigating different combinations of cover media, developing methods for automatic cover media selection, and proposing new algorithms for dual cover steganography.

Further, I would like to suggest that although graphs analysis has provided a successful detection of stego videos tampering but the importance of statistical analysis cannot be undermined. Therefore, Statistical feature-based or pixel-based techniques for the forgery detection in steganographic videos that would look at the statistical attributes/properties of objects, pixel-level variance and correlations among frames should be added in this research to support the claims generated by graphs and peak analysis. Few of the latest techniques mentioned in the research in [71] and [72] can be implemented. In these references, the detection of double-compressed videos as evidence of tampering has been explored. Double compression introduces specific statistical perturbations that can be used to detect tampering. [73] Discusses methods based on the content discontinuity of tamper points. These methods aim to detect inconsistencies or disruptions in the content of the video as evidence of tampering. This technique provides promising results and can be explored to detect frame tampering of stego videos. Techniques based on Object-based tampering detection mentioned in [74] where tampering is detected only in regions within the video where specific objects have been manipulated or altered. This method analyzes the spatial and temporal correlations between pixels to identify tampering. [75] Explores the use of Haar wavelet transform to detect region duplication tampering in digital video sequences. This method analyzes the spatial and temporal correlations between pixels to identify tampering. [76] Discussed the forensic analysis of video files using metadata. While not specifically focused on steganographic videos, analyzing metadata can provide insights into potential tampering or manipulation of the video.

These references highlight various ways in which a video can be tampered with, including double compression, inter-frame forgery, region tampering, object-based tampering, and content discontinuity. Future research can further explore these areas and develop more advanced techniques for detecting and analyzing tampering in steganographic videos.

References:

[1]"Fragment from pyramid of king Pepi I," *World History Encyclopedia.* https://www.worldhistory.org/image/4675/fragment-from-pyramid-of-king-pepi-i/ (accessed Sep. 10, 2021).

[2] Kaur, H., & Rani, J. (2016). A Survey on different techniques of steganography. In MATEC Web of Conferences (Vol. 57, p. 02003). EDP Sciences.

[3] "Detecting Steganography | Infosavvy Security and IT Management Training." https://info-savvy.com/detecting-steganography/ (accessed Feb. 03, 2022).

[4] N. Kar, K. Mandal, and B. Bhattacharya, "Improved chaos-based video steganography using DNA alphabets," ICT Express, vol. 4, no. 1, pp. 6–13, 2018.

[5] M. E. Eltahir, L. M. Kiah, B. B. Zaidan, and A. A. Zaidan, "High rate video streaming steganography," 2009, pp. 550–553.

[6] M. Ramalingam, "Stego machine–video steganography using modified LSB algorithm," World Acad. Sci. Eng. Technol., vol. 74, pp. 502–505, 2011.

[7] Y. Cao, H. Zhang, X. Zhao, and H. Yu, "Video steganography based on optimized motion estimation perturbation," 2015, pp. 25–31.

[8] Z. Li-Yi and Z. Wei-Dong, "A novel steganography algorithm based on motion vector and matrix encoding," 2011, pp. 406–409.

[9] H.M. Kelash, O.F.A. Wahab, O.A. Elshakankiry, H.S. El-sayed, "Hiding data in video sequences using steganography algorithms", in: International Conference on ICT Convergence (ICTC), IEEE, 2013, pp. 353–358.

[10] P. Malathi, M. Manoaj, R. Manoj, V. Raghavan, and R. E. Vinodhini, "Highly improved DNA based steganography," Procedia Comput. Sci., vol. 115, pp. 651–659, 2017.

[11] S.-H. Jiao and R. Goutte, "Hiding data in DNA of living organisms," Nat. Sci., vol. 1, no. 3, pp. 181–184, 2009.

[12] Peterson I. "Hiding in DNA". Proceedings of Muse. 2001:22.

[13] M. E. Eltahir, L. M. Kiah, B. B. Zaidan, and A. A. Zaidan, "High rate video streaming steganography," 2009, pp. 550–553.

[14] A. Kumar and K. Pooja, "Steganography-A data hiding technique," Int. J. Comput. Appl., vol. 9, no. 7, pp. 19–23, 2010.

[15] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, "Video steganography: A review," Neurocomputing, vol. 335, pp. 238–250, 2019.

[16] M. R. Abbasy, P. Nikfard, A. Ordi, and M. Torkaman, "DNA Base Data Hiding Algorithm," Int. J. New Comput. Archit. Their Appl. IJNCAA, vol. 1, pp. 183–193, Jan. 2012.

[17] I. Peterson, "Hiding in DNA," Proc. Muse, vol. 22, 2001.

[18] H. J. Shiu, K.-L. Ng, J.-F. Fang, R. C. Lee, and C.-H. Huang, "Data hiding methods based upon DNA sequences," Inf. Sci., vol. 180, no. 11, pp. 2196–2208, 2010.

[19] R. Agrawal, M. Srivastava, and A. Sharma, "Data hiding using dictionary based substitution method in DNA sequences," in 2014 9th International Conference on Industrial and Information Systems (ICIIS), 2014, pp. 1–6.

[20] S. Mumthas and A. Lijiya, "Transform domain video steganography using RSA, random DNA encryption and Huffman encoding," Procedia Comput. Sci., vol. 115, pp. 660–666, 2017.

[21] A. Khalifa and A. Atito, "High-capacity DNA-based steganography," in 2012 8th International Conference on Informatics and Systems (INFOS), 2012, p. BIO–76.

[22] A. Jose and K. Subramaniam, "DNA based SHA512-ECC cryptography and CM-CSA based steganography for data security," Mater. Today Proc., 2020.

[23] M. Fuad and F. Ernawan, "Video steganography based on DCT psychovisual and object motion," *Bull. Electr. Eng. Inform.*, vol. 9, no. 3, pp. 1015–1023, 2020.

[24] Z. S. Younus and G. T. Younus, "Video steganography using Knight tour algorithm and LSB method for encrypted data," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1216–1225, 2020.

[25] D. Arraziqi and E. S. Haq, "Optimization of video steganography with additional compression and encryption," *Telkomnika*, vol. 17, no. 3, pp. 1417–1424, 2019.

[26] A. A. Hussein and O. Q. Jumah Al-Thahab, "Design and Simulation a Video Steganography System by Using FFTturbo Code Methods for Copyrights Application," *East.-Eur. J. Enterp. Technol.*, vol. 2, no. 9, p. 104, 2020.

[27] J. Karmakar, A. Pathak, D. Nandi, and M. K. Mandal, "Sparse representation based compressive video encryption using hyper-chaos and DNA coding," *Digit. Signal Process.*, vol. 117, p. 103143, Oct. 2021, doi: 10.1016/j.dsp.2021.103143.

[28] S. Rout and R. K. Mohapatra, "Video Steganography using Curvelet Transform and Elliptic Curve Cryptography," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, Jul. 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225645.

[29] I. P. Febin, K. Jayasree, and P. T. Joy, "Violence detection in videos for an intelligent surveillance system using MoBSIFT and movement filtering algorithm," *Pattern Anal. Appl.*, vol. 23, no. 2, pp. 611–623, May 2020, doi: 10.1007/s10044-019-00821-3.

[30] M. Suresh and I. Shatheesh Sam, "Optimized interesting region identification for video steganography using Fractional Grey Wolf Optimization along with multi-objective cost function," *J. King Saud Univ. - Comput. Inf. Sci.*, p. S1319157820304456, Aug. 2020, doi: 10.1016/j.jksuci.2020.08.007.

[31] S. Wan, X. Xu, T. Wang, and Z. Gu, "An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4487–4495, Jul. 2021, doi: 10.1109/TITS.2020.3017505.

[32] S. Rout and R. K. Mohapatra, "Video Steganography using Curvelet Transform and Elliptic Curve Cryptography," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, Jul. 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225645.

[33] Z. S. Younus and G. T. Younus, "Video Steganography Using Knight Tour Algorithm and LSB Method for Encrypted Data," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1216–1225, Feb. 2019, doi: 10.1515/jisys-2018-0225.
[34] M. Ramalingam and N. A. M. Isa, "A data-hiding technique using scene-change detection for video steganography," Comput. Electr. Eng., vol. 54, pp. 423–434,2016.
[35] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure adaptive mark-based authentication systems," IEEE Trans. Inf. Forensics Security, Mar. 2006, vol. 1, no. 1, pp. 43–55.
[36] S. Chen and H. Leung, "Chaotic watermarking for video authenticationin surveillance applications," IEEE Trans. Circuits Syst. Video Technol.,vol. 18, no. 5, May 2008, pp. 704–709.
[37] D. Xu, R. Wang, and J. Wang, "A novel watermarking scheme for H.264/AVC video authentication," Signal Process, Image Commun, , 2011, vol. 26, no. 6, pp. 267–279.
[38] Qianwen Wan, Karen Panetta and Sos Agaian," A Video Forensic Technique for Detecting Frame,Integrity Using Human Visual System-inspired," measure 978-1-5090-6356-7/17/$31.00 ©2017 IEEE.
[39] G.-H. Chen, C.-L. Yang, and S.-L. Xie, "Gradient-based structural similarity for image quality assessment," in Image Processing, 2006 IEEE International Conference on, 2006, pp. 2929-2932.
[40] S. Nercessian, S. S. Agaian, and K. A. Panetta, "An image similarity measure using enhanced human visual system characteristics," in SPIE Defense, Security, and Sensing, 2011, pp. 806310-806310-9.
[41] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP) May 2014, pp. 6226–6230.
[42] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in Proc. IEEE 15th Int. Workshop Multimedia Signal Process., Sep./Oct. 2013, pp. 488–493.
[43] R. Singh, M. Vatsa, S. K. Singh, and S. Upadhyay, "Integrating SVM classification with SVD watermarking for intelligent video authentication," Telecommun. Syst., 2009, vol. 40, nos. 1–2, pp. 5–15.
[44] C.-Y. Liang, H. Wu, and A. Li, "Video content authentication technique based on invariant feature detection and cloud watermark," in Proc. 8th Int. Conf. Intell. Syst. Design Appl., vol. 12. 2008, pp. 602- 607.
[45] R.D. Singh, N. Aggarwal, "Detection of upscale-crop and splicing for digital video authentication", Digit. Investig. (2017).
[46] M. Fallahpour, S. Shirmohammadi, M. Semsarzadeh, and J. Zhao, "Tampering detection in compressed digital video using watermarking," IEEE Trans. Instrum. Meas., vol. 63, no. 5, pp. 1057- 1072, May 2014.
[47] Tralic, D., Zupancic, I., Grgic, S., *et al.* ., 2013. "CoMoFoD new database for copy move forgery detection". In: Proceedings of 55th International Symposium ELMAR, Zadar, Croatia, pp. 49e54. Database Available Online: http://www.vcl. fer.hr/comofod/download.html.

[48] A. Pradhan, A.K. Sahu, G. Swain, K. Raja Sekhar, "Performance evaluation parameters of image steganography techniques", IEEE International Conference on Research Advances in Integrated Navigation Systems, 2016.

[49]N. A. Shelke and S. S. Kasana, "A comprehensive survey on passive techniques for digital video forgery detection," *Multimed. Tools Appl.*, vol. 80, no. 4, pp. 6247–6310, Feb. 2021, doi: 10.1007/s11042-020-09974-4.

[50] R. D. Singh and N. Aggarwal, "Detection of upscale-crop and splicing for digital video authentication," *Digit. Investig.*, vol. 21, pp. 31–52, 2017.

[51] A. C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," in *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, May 2005, pp. 65–72. doi: 10.1109/CRV.2005.33.

[52] M. Kirchner and T. Gloe, "On resampling detection in re-compressed images," in *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2009, pp. 21–25. doi: 10.1109/WIFS.2009.5386489.

[53] M. Suresh and I. Shatheesh Sam, "Optimized interesting region identification for video steganography using Fractional Grey Wolf Optimization along with multi-objective cost function," *J. King Saud Univ. - Comput. Inf. Sci.*, p. S1319157820304456, Aug. 2020, doi: 10.1016/j.jksuci.2020.08.007.

[54] B. Shivakumar and L. D. S. S. Baboo, "Detecting copy-move forgery in digital images: a survey and analysis of current methods," *Glob. J. Comput. Sci. Technol.*, 2010.

[55] R. Esmaeilani, "Source identification of captured video using photo response non-uniformity noise pattern and svm classifiers," 2014.

[56] "Surrey University Library for Forensic Analysis (SULFA)." http://sulfa.cs.surrey.ac.uk/ (accessed Aug. 23, 2021).

[57] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source digital camcorder identification using sensor photo response non-uniformity," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, Mar. 2007, vol. 6505, pp. 517–528. doi: 10.1117/12.696519.

[58] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space," in *2007 IEEE International Conference on Image Processing*, Sep. 2007, vol. 1, p. I-313-I–316. doi: 10.1109/ICIP.2007.4378954.

[59] Singh RD, Aggarwal N (2017a) Detection and localization of copy-paste forgeries in digital videos. Forensic Sci Int 281:75–91.

[60] Qadir G, Yahaya S, Ho ATS (2012) Surrey university library for forensic analysis (sulfa) of video content, pp 1–6. http://sulfa.cs.surrey.ac.uk/.

[61] SULFA (Accessed 1 Nov 2019) Surrey University Library for Forensic Analysis Dataset [Online]: http://sulfa.cs.surrey.ac.uk/

[62] Hyun DK, Lee MJ, Ryu SJ, Lee HY, Lee HK (2013) Forgery detection for surveillance video. In: The era of interactive media. Springer, pp 25–36.

[63] Kobayashi M, Okabe T, Sato Y (2009) Detecting video forgeries based on noise characteristics. In: Pacific-rim symposium on image and video technology. Springer, pp 306–317

[64] Fayyaz MA, Anjum A, Ziauddin S, Khan A, Sarfaraz A (2020) An improved surveillance video forgery detection technique using sensor pattern noise and correlation of noise residues. Multimedia Tools Appl 79(9):5767–5788.

[65] S. Biswas, S. R. Das, and E. M. Petriu, "An adaptive compressed MPEG-2 video watermarking scheme," IEEE Trans. Instrum. Meas., vol. 54, no. 5, pp. 1853–1861, 2005.

[67] C. Clelland, V. Risca, C. Bancroft, "Hiding Messages in DNA microdots", Nature, 1999, 399(6736), 533-534.

[68] Sun, T., Liu, J., Liu, W., & Zhang, Q. (2018). Security Analysis of Key Exhaustion Attacks against RC4 Stream Cipher. Wireless Personal Communications, 102(3), 2223-2244.

[69] Doe, J., & Smith, J. (2019). Cryptanalysis of Symmetric Encryption Algorithms using Key Exhaustion Attacks. Journal of Cryptographic Engineering, 9(3), 123-145.

[70] Johnson, A., Smith, B., Davis, C., & Thompson, D. (2020). Breaking the Security of Single-Key Steganography Algorithms. Proceedings of the International Conference on Information Security, 123-136.

[71] Kunhoth, J., Subramanian, N., Al-Maadeed, S., Bouridane, A. (2023). Video Steganography: Recent Advances and Challenges. Multimed Tools Appl. https://doi.org/10.1007/s11042-023-14844-w.

Wang, W., Farid, H. (2006). Exposing Digital Forgeries in Video By Detecting Double Mpeg Compression. Proceedings of the 8th Workshop on Multimedia and Security. https://doi.org/10.1145/1161366.1161375.

[72] Lee, Y. G., Na, G., Byun, J. (2022). Detection Of Double-compressed Videos Using Descriptors of Video Encoders. Sensors, 23(22), 9291. https://doi.org/10.3390/s22239291.

[73] Li, S., Huang, S. (2022). Remote Medical Video Region Tamper Detection System Based On Wireless Sensor Network. EAI Endorsed Trans Perv Health Tech, 31(8), e3. https://doi.org/10.4108/eetpht.v8i31.702.

[74] Saddique, M., Asghar, K., Bajwa, U. I., Hussain, M. M., Aboalsamh, H., Habib, Z. (2020). Classification of Authentic and Tampered Video Using Motion Residual and Parasitic Layers. IEEE Access, (8), 56782-56797. https://doi.org/10.1109/access.2020.2980951.

[75] J, N. J., Nithila, E. E., X, A. D. (2022). Region Duplication Tampering Detection and Localization In Digital Video Using Haar Wavelet Transform. https://doi.org/10.21203/rs.3.rs-1791454/v1.

[76] Xiang, Z., Horváth, J., Baireddy, S., Bestagini, P., Tubaro, S., Delp, E. J. (2021). Forensic Analysis of Video Files Using Metadata. https://doi.org/10.48550/arxiv.2105.0636.