



Towards Video Secure Streaming- Feasibility Study of Using an Obscuring Algorithm in Conjunction of H.264 Encoding and Compression

**DEEPIKA CHALLA
SURYA TEJA VULAVAKAYALA**

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Telecommunications Engineering. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:

Author(s):

Deepika Challa

E-mail: dech18@student.bth.se

Surya Teja Vulavakayala

E-mail: suvu18@student.bth.se

University advisor:

Dr. Siamak Khatibi

Department of Technology and Aesthetics

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet: www.bth.se
Phone: +46 455 38 50 00
Fax: +46 455 38 50 57

ABSTRACT

Technology advancement increases the usage of the internet day by day. One of the most used internet services is video streaming. The major advantage of video streaming is that it allows long distance communication between people without any delay. It is known that streaming video is one of the fastest growing industries, and it has been very beneficial to the world. As the use of video streaming is increasing rapidly, it is essential to have security for video streaming. A lot of methods to secure video streaming came into existence like authentication, protocol, or some secure web hosting sites. Every method is to secure the transmission of video streaming, so these methods use either key or any other additional things to secure it. Our approach is to have a secure video streaming method without using any additional key or software. So, the method here is to encrypt the video directly and then encode it to be in the streaming format. The encryption of the video is done by a method called obscuring method. And the performance evaluation is done to the method so as to check the stability and feasibility of the system.

Keywords: H.264 codecs, H.264, video streaming codecs, Security in video streaming, Security methods in video streaming, Video streaming

ACKNOWLEDGMENT

*Firstly, we feel very grateful and honoured to do our thesis under the supervision of **Dr. Siamak Khatibi**. It was out of the question to do our thesis without his guidance and support. It was also a huge help for us to meet with him on a weekly basis during the thesis process. He taught us how to write our thesis, which helped us complete it. He helped us a lot in difficult situations, he encouraged us a lot in our work, and his suggestions also helped us a lot. He was the person who graciously allowed us to carry out this thesis successfully. He has been invaluable in guiding our progress and always made himself available if we had any questions or difficulties. We are very grateful that he offered his supervision and support throughout this process.*

We are glad to have our family and friends' support and unconditional love to make our thesis prosperous.

Contents

Abstract	ii
Acknowledgment	iii
List of Figures.....	vi
List of Tables.....	vii
List of Abbreviations	vii
1. Introduction	1 - 5
1.1 Streaming.....	1
1.2 Video streaming and its importance	1
1.3 Security issues in video streaming.....	1
1.4 Solutions introduced to secure the video streaming.....	2
1.5 Motivation.....	3
1.6 Aim and Objectives.....	3
1.7 Research Questions.....	4
1.8 Outline	4
1.9 Contribution to thesis.....	5
2. Related work.....	6 - 9
2.1 Research Methodology	6
2.1.1 Literature Review	6
2.1.2 Search Strategy	6
2.1.3 Inclusive Criteria and Exclusive Criteria.....	6
2.1.4 Data Extraction	7
2.2 Related work.....	7
2.2.1 Existing algorithm.....	7
2.2.2 List of video codecs	8
2.2.3 Comparison of video codecs.....	8
2.2.4 Papers related to H.264.....	9
3. Method	10 - 14
3.1 Streaming process	10
3.2 Encoding	10
3.3 Storage	12
3.4 Delivery	13
3.5 H.264 Overview & working	14
4. Experimentation	15 - 30
4.1 Encoding process	15
4.1.1 First part of encoding process.....	17

4.1.2	Color image processing	17
4.1.3	Obscuring process	18
4.2	Second part of encoding process.....	20
4.2.1	Motion estimation	20
4.2.1.1	Novel four-step search algorithm	21
4.2.2	Motion compensation	22
4.2.2.1	Variable block motion compensation	23
4.2.3	Intra frame prediction	25
4.2.4	Image compensation model	26
4.2.4.1	Discrete Courier Transform	27
4.2.4.2	Quantization.....	27
4.2.4.3	Entropy encoder.....	27
4.3	Decoding part.....	27
4.4	Materials	29
5.	Result and Analysis.....	31 - 41
5.1	Encoding and decoding results	31
5.2	Scaling factor	32
5.3	Texture factor.....	37
6.	Discussion.....	42 – 44
7.	Conclusion and Future work	45
	References	46

List of figures

3.1	Clear view of streaming process	10
3.2	Video codecs with their organizations and years	11
3.3	Information saved in file containers	12
3.4	Process of video streaming	13
4.1	Flowchart of encoding process	15
4.2	Explanation of frames and their blocks	16
4.3	First step in encoding process	17
4.4	Color image processing from RGB to YCbCr	18
4.5	Frame division input blocks of 32-bit size	18
4.6	Splitting of block into pieces	19
4.7	Block pieces arranged into Tree encryption method	20
4.8	Second step in encoding process	20
4.9	Comparison of macro block in search area between 2 frames	21
4.10	4ss step search algorithm	22
4.11	Reference frame and motion compensated frame	23
4.12	FBMC and VBMC	24
4.13	Matching of macro block with the Root macro block	24
4.14	Macro block sizes w.r.t motion	25
4.15	Intra mode predictions of all directions	26
4.16	Frame going through the image compression model steps	26
4.17	Block pieces into block with Tree decryption method	28
4.18	First three subsequent frames of video-1	29
4.19	First three subsequent frames of video-2	30
4.20	First three subsequent frames of video-3	30
4.21	First three subsequent frames of video-4	30
5.1	Results of video-1	31
5.2	Video-1 with original pixel resolution	33
5.3	Video-1 with low pixel resolution	35
5.4	Video-1 with High pixel resolution	36
5.5	Video-2 with original pixel resolution	38
5.6	Video-3 with original pixel resolution	39
5.7	Video-4 with original pixel resolution	41
6.1	Noise comparison of different pixel resolutions	43
6.2	Encrypted frames comparison of different pixel resolutions	43

List of tables

1.1	Contribution table	5
4.1	Properties of different videos.....	29

List of abbreviations

ISO	International Standard Organization
VCEG	Video Coding Experts Group
ITU-T	International Telecommunications union
MPEG	Motion Picture Experts Group
MP4	MPEG layer4
AVI	Audio Video Interleave
FLV	Flash video
MKV	Matroska Multimedia Container
WMV	Windows Media Video
WebM	Web Method
RTMP	Real-Time Messaging Protocol
RTSP	Real-Time Streaming Protocol
HLS	HTTP Live Streaming
URL	Uniform Resource Locator
MAD	Mean Absolute Difference
SSD	Sum of Squared Difference
VCL	Video coding Layer
VOD	Video On Demand
AVC	Advanced Video Coding
DCT	Discrete Cosine Transform
RGB	Red Green Blue
DDOS	Distributed Denial Of Service
VBMC	Variable Block Motion Compensation
VLC	Video LAN Client
JPEG	Joint Photographic Experts Group
DPCM	Differential Pulse Code Modulation
HVS	Human Visual System

MSE	Mean Square Error
DRM	Digital Rights Management
SAD	Sum of Absolute Difference
CCITT	Commission Consultative International Telegraphique et Telephonique
IEC	International Electrotechnical Commission
AAC	Advanced Audio Coding
RLE	Run Length Coder
FBMC	Fixed Block Motion Compensation
AES	Advanced Encryption Standards
SAE	Sum of Absolute Error
HVEC	High Efficiency Video Coding
PSNR	Pulse Signal to Noise Ratio
ECC	Elliptic Curve Cryptography

1.1 Streaming

As the use of the internet globally is increasing from day-to-day life, streaming is one of the most used internet services[1][2]. Streaming is nothing but the media that transmits and receives the data over the internet. Playback can be started before the whole file is downloaded. Streaming is more like a pre-loading process that starts as soon as the player loads. It is an efficient way than downloading and storing files on your device. Video, audio, image, data are streaming media that can be delivered over any streaming service. Streaming can be categorized into two categories: live and on-demand streaming. Live streaming refers to an event or show that is broadcast online with no delay in between, while on-demand streaming refers to content that is hosted online without any specific time or date set, unlike live streamed content which streams online at a specified time and date[29].

1.2 Video streaming and its importance

Out of all, video streaming is one that drains most of the internet when compared with other streaming media and with a lot of demand[3][4]. Video is a assemble of images with a sequence of audio plays along with the images. A stream of these images, when transmitted over the internet to the player in video form, is known as video streaming. The player then combines the audio with the images to make it a video and then plays it on the computer. The main purpose of video streaming is to transmit a lot of data over the network, but with low latency and reliable performance. Sites such as YouTube and Twitch broadcasts live streaming of events such as news, sports, and entertainment on their websites or applications.

Video streaming is becoming more prevalent in the professional world. It helps organizations connect with their audience in a more personal and intimate way. Video streaming is used in a wide variety of contexts such as online classes[28], service promotions, gaming, and many more. As some of the video streaming services offer free of cost, it made some of the users to watch video streams. Video streaming made storage free devices for viewers, as no not to download the video to watch.

1.3 Security issues in video streaming

According to the Pew Research Center study in 2019, more than half of Millennials are already multi-screening. That's just one of many reasons why video streaming has become so popular in the last decade. But with technological advancements, video streaming is becoming increasingly vulnerable to security issues[14] and threats such as DDoS attacks, malware infections, and copyright infringement. A keen overview of security issues is explained in video streaming[27]. Some types of security issues are explained below.

A distributed denial of service (DDoS) attack is a cyberattack, usually carried out by an organized group of people or bots (a robot or software) on multiple targets at one time. The attack works by flooding the bandwidth or resources of a targeted system with superfluous requests[43]. That makes a legitimate request impossible to complete, effectively denying service to legitimate users. An example of a DDoS attack would be when someone plays an annoying video on a popular streaming site and thousands of people simultaneously try to access it, which causes the server to crash or slows it down significantly due to the high volume of traffic. A DDoS attack can target any computer connected to the Internet and is not restricted by borders.

Malware infections are rarely encountered on streaming sites, but malware designed to infect computers that visit streaming websites is common[26]. The majority of malware infections on these sites are through malicious plugins, especially Adobe Flash Player and Java, which users sometimes unwittingly install when they click an advertisement or download a new software without knowing what it is.

The most common type of online copyright infringement[44] is unauthorized downloading. Most people who watch pirated movies and shows from illegal streaming websites do so because they don't want to pay for the content or because it's available for free.

As these are some of the security issues explained, a lot of them are causing attacks on streaming video. For these types of attacks, a lot of techniques came into existence to avoid them. These techniques are discussed in the following section.

1.4 Solutions introduced to secure the video streaming

When the data is streamed over the internet, the main risk that arises is the security of the data. The data can be copied and used in some or the other way, or the important data can be stolen. Here data refers to video. There are so many ways to secure the video while streaming. Securing the video means prioritizing the security and privacy of the video. Some of the security issues are piracy and hacking. So due to these concerns, there are some solutions that came into existence like DRM, encryption techniques, protocols, web servers, and so on.

DRM (Digital Rights Management) is one of the best ways to secure video while streaming[5]. DRM is used to prevent copyright infringement[46]. This encryption has a decryption key that enables us to decrypt the encrypted video before watching it on an authorized device. The device should be authorized by DRM and only then it will decode the encrypted video. The decryption key can be stolen or copied, and it can be used to playback the malicious code thus corrupting the system.

Encryption techniques like RC4 are used to avoid piracy and hacking. By encrypting the video before streaming, we can avoid any unauthorized access to our valuable data. The encryption will be done before streaming and the decryption will be done on the client side by using a decryption key. RC4 is an encryption technique that uses the key for encrypting and decrypting the data. It uses a random number generator for generating a 128-bit key for encryption or decryption[6]. HLS encryption, with AES 128 bit encrypted streaming, is used to secure the video streaming for Netflix, Vimeo and other such websites[45]. This way there are many techniques to secure the video stream.

Even some protocols came into existence to secure the video streaming. HTTPS ensures that data is not tampered with during transit and encrypts the connection to prevent eavesdropping and man-in-the-middle attacks. It usually offers a higher quality stream than HTTP, but this depends on the implementation of the protocol and can be degraded by certain objects such as proxies that may be required to unblock controversial content. There are some other protocols too, for securing the video streaming. This paper introduces[42] a protocol that explains its implementation of it and some drawbacks are there to the protocol. This protocol secures the video transmission too.

When coming to security solutions for video streaming, there are lots of solutions to secure it. A key or any other encryption standard is used to secure the video that is streamed. Here our main idea is not to use any key or any other encryption or software for security. But this thesis shows a very new approach to secure the streaming video by directly encrypting the video itself. So even if the video got into the hands of hackers, they don't even have an idea of decrypting the video without knowing the encryption technique. The video encryption is done by a unique method called obscuring method. Obscuring method is to conceal or hide any data as if it's totally covered and not even visible. This way, we can secure the video directly, and this method can be helpful to many sectors like the army, medical, and so on. Here the main question is how exactly the video can be encrypted.

1.5 Motivation

The motive behind this thesis is to have an encrypted video be streamed over the internet instead of using all the other methods to secure the stream. As even this video is copied or stolen by a hacker, it can't be retrieved back as the encryption method is unknown. And to know which method is used is totally not an easy mission to do. The encrypted video has lots of uses in lots of sectors like the army, medical, and so on. To the fields that want to send the data safer, this method can be used without any hesitation. And this idea of having encrypted video has lots of scope for security in the future. This video encryption method neither uses any key at the decryption side, nor any other software is used.

1.6 Aim and objectives

Aim

Our main aim is to encrypt the video while it is being encoded. This encryption can be done by obscuring method. So, while the video is in the process of encoding, the process is interrupted in between, and the encryption of the video is done, and let the rest of the encoding process is done as it is. By encrypting the video, even if it gets hacked, it is not an easy mission to retrieve the information in the video back as the encryption technique is unknown. So, this encryption is done by an algorithm called obscuring method. Our idea is to have the video to be obscured(hide), as the video can be divided into frames. So, these frames can be encrypted by using an obscuring method. The main question here is whether an encrypted video can be encoded or not. For that, we need to go through all the streaming process steps explained in the 3rd chapter. Then the process of this starts by selecting a codec to use in order to complete the

first step of steaming. Then we need to check how the codec works by sending an encrypted video over the codec. After that, a performance evaluation should be done by having different video files with different patterns in video and different pixel ratios.

Objectives

- To find a codec that is suitable to transmit the encrypted video.
- To investigate the possibility of having the obscuring method by interrupting the codec.
- To find a way of encrypting all the frames of the video using the obscuring method.
- To investigate the performance evaluation of the codec by using different video clips and check its working of it.

1.7 Research questions

RQ1: what is the feasibility of the system while using the obscuring method?

RQ2: How stable is the system having the obscure video in the encoded format?

1.8 Outline

Basically, this thesis draft consists of seven chapters. The first chapter discusses streaming and its importance, followed by security concerns and existing technologies to secure video streaming are discussed. There after motivation was explained and the aim, objectives, and research questions are listed, and at the end of this chapter, we have mentioned the contribution of the thesis. In addition to the second chapter, we clearly explained the method used for the literature search, then an overview of the literature is given in the related work regarding the research and implementation of streaming. Furthermore, in the third chapter, we explained the methodology that shows the reader a view of what exactly this thesis is about. It starts with the streaming process and steps. Then in the fourth chapter, we start to explain the process to implement the video encryption and then encoding process. The encoding process is explained in two parts. In this section, the reader can have a clear view on the process used. And then, the decoding part is explained. By the end of this chapter, we have provided the materials that are used for the experiment and the properties of the videos are listed in tabular form. In chapter 5, we acquired the results of video that is encrypted and encoded into streaming format. The results are to declare that this encoding of encrypted video is possible, and the performance of the system is evaluated by using the videos from the material section. In chapter 6, we discussed the process and our results which are acquired from the experimentation. Then the research questions are answered. Finally, in chapter 7, we end our thesis with a conclusion and future work of the whole research. By the end of this section, the reader gets a clear view of our thesis.

1.9 Contribution to thesis

The thesis work is divided based on the strength and weaknesses of the author. The entire work includes theoretical research, simulation, and analysis of the results. The work is shared by both authors equally based on the topics, and all the major topics cover the main parts of the thesis. The simulation is done in MATLAB. Most of the works are covered by both authors because the thesis contains more work. The below mentioned table represents the topics and contributions made by each author (Table 1.1)

Topic	Author 1 - Deepika	Author 2 - Surya Teja
Introduction	✓	
Research Methodology	✓	
Background study	✓	✓
Aim and Objectives		✓
Research questions	✓	✓
Outline	✓	
Methodology	✓	✓
Experimentation	✓	✓
Result and Analysis	✓	✓
Discussion	✓	✓
Conclusion and Future work	✓	✓

Table 1.1: Contribution to thesis

2.1 Research methodology

2.1.1 Literature review

We would like to begin our literature search to know the easiest way to create an environment for video streaming. Firstly, we would like to study if there is any appropriate information regarding the secure video streaming parameters. To find the papers, we used a Boolean search like “H.264 video Codecs” or “video streaming.” After getting the results of the Boolean search, we select the papers by using inclusion and exclusion criteria.

2.1.2 Search strategy

Firstly, we start our research on finding relevant papers by searching in IEEE, ACM Library, Google Scholar, Scopus. Here we search the papers by using keywords like video streaming, video codec, video formats. From this search, we extracted 100 papers on the basis of title, and a total of 75 papers are extracted based on content, and then from these 75 papers, finally we select 55 papers based on abstract and the topics which are related to our thesis.

2.1.3 Inclusion and exclusion criteria

Inclusive criteria

- Papers that are published in English
- Papers with the clear aim
- Papers that are discussed regarding video streaming
- Papers that discussed security in video streaming
- Papers that are presented, improved, and customized for security in video streaming.

Exclusive criteria

- Papers that are not published in English
- Papers without a clear aim
- Papers that are not discussed about video streaming.
- Papers that are not discussed about security in Video streaming
- Papers that are not mentioned in streaming

2.1.4 Data extraction

From the selected papers, suitable data is extracted. From primary studies, the research questions are examined. The information that we extracted is used for the documentation to explain our thesis clearly. The information extracted from each paper includes the following:

paper ID, paper title, publication year, publication type, domain.

2.2 Related work

This section represents the related work for this thesis. The papers that are provided in this section are related to the topics of existing algorithms to secure the video streaming, the list of video codecs, the comparison of video codecs and the selected codec related papers.

2.2.1 Existing algorithms

The papers listed in this section are the existing security methods for video streaming. The papers show the reason behind the method this thesis used.

This author [12] explains different security methods to secure real-time video streaming with a key, and the security is done in the application layer. The key is used to generate the data blocks that are transposed in between the video packets. For example, if there are a few data blocks in a packet with some sequence, the key generates a new sequence. This sequence can't be played without the secret key, which is transmitted through a secure channel. Here a key is used to secure the video stream.

This paper[10] discusses a security approach regarding video streaming by using the SRTP protocol. This encryption uses AES-CFB video streams to secure transmission through the client-server network while using H.264 as a video compression technique. Also, they use the AES encryption technique to encrypt the streaming video. But in this encryption, there are multiple level encryption techniques used as the authentication of client-side and server-side password encryption, URL video encryption, video file encryption. So, to have a secure streaming system, this paper also did some password cracking tests and some more to see whether it is easy to decrypt or not.

In this paper[13], the author explains a novel idea that has advantage with the two methods in securing the video. The author firstly uses a hybrid technique encryption by using AES and ECC to encrypt the video with the help of the key. Here the author shows the key as a main factor in encryption technique used. secondly the author done a comparison between the proposed technique with the existing technology and the results shows that the proposed technique has better outcomes when compared.

This paper[47] has a different approach to securing the video stream in a wireless network through a new SSS (Secure Scalable Streaming) method. This approach aims for scalability, efficiency, and security of the video. This SSS method firstly encodes the given video into secure and scalable packets. These packets are streamed through a wired or wireless to clients over heterogeneous networks. This method then performs the transcoding of packets without decrypting them.

There are more methods that came into existence to secure the video streaming. Each has its advantages and different type of methods. So the thing which we observed in every existing algorithm is it uses an additional substitute like a key or any other additional things to secure the video. For that, we have selected a method called a recursive method that does even use any key or any other aids and converts the image into noise for security[37]. So this method is used for this thesis implementation.

2.2.2 List of Video codecs

Here we have given an overview on all the codecs that are introduced and also explained. From

these codecs, thesis uses a video codec, and given a reason in selecting one particular codec is explained. Here the listed codecs are existing ones that are in use.

In this paper [48] author explains the AV1 video codec. This codec was developed early in 2018 by the alliance for open media industry consortium. The main aim of this paper was to develop an AV1 codec to achieve hardware feasibility and compression complexity. This paper clearly explains the coding tools and used coding techniques. This paper conducted a comparison between the VP9 video codec and AV1 to show which performs better.

The paper [49] is about a VP9 codec which is an open source video codec for new generations. It was introduced by Google. This paper explains an overview of VP9 and the coding tools like prediction block sizes, prediction modes, and many more are clearly explained. This paper finally compares with the other codecs using the PSNR sense. They used the Bjøntegaard Delta Rate metric to compare the codecs.

The paper [50] explains different concealment techniques for spatial and temporal errors in MPEG-2. This paper gives a little explanation regarding the codec, and then error detection techniques are explained clearly. After finding out the error by these techniques, they have used error concealment for spatial and temporal of motion compensation and motion estimation. The author also experimented after the concealment, using PSNR.

The author of this paper [51] clearly explains the HVEC (High Efficiency Video Coding) codec. The author shows the importance of this codec, key features, and some highlights. Then continues to give an overview of the working of this codec, including the coding techniques used in this HVEC codec. The author provides a brief explanation of everything that includes in the codec.

These are a few of the codecs that exist today. Here this thesis is just not going for the better codec or best quality provided codec. This thesis is just to check whether there is a possibility of encoding the encrypted video by obscuring method. Here this thesis focuses on H.264 as it is the most widely used codec among the existing codecs. And H.264 is compatible with almost all devices.

2.2.3 Comparison of Video Codecs

The comparison between the codecs is done, and every paper shows that H.264 codec is better than all the other codecs. Paper that compares the codecs on which factors are also explained clearly in those papers which are listed below. These papers also explain about H.264 codec.

The paper[8], H.264/AVC this paper explains Fidelity Range Extensions (FRExt) that are introduced. It also explains the features of these extensions and some highlights are also provided. This paper shows what exactly those extensions are and provides a vast explanation of these extensions. After all these, the author compared this H.264 codec with the existing codec that is MPEG-2. The performance of the H.264 codec is high when compared with the MPEG-2.

In this paper[16], the author conducts a performance evaluation on different video codecs. Firstly, the author starts with the video quality metric and PSNR ratio introduction. And using these two metrics, the comparison of codecs is done between the H.264 codec, MPEG-2 video codec, MPEG-4 video codec, and H.263. These codecs are compared with different input files and H.264 stood first in this comparison.

The authors[17] explain the process which was done in codecs and how the codec works, and then they compare the pros and cons of each codec and also the bitrates and time rates are compared. The comparison of codecs is done between the M-JPEG, MPEG-4 part 2, and H.264. The results show that H.264 is the best of the rest of them.

This author[18] clearly explains the in-depth concepts of the H.264 video codec. Each and every step in this codec is compared with the other codecs and explains the positives of H.264. Finally, the authors end with the performance evaluation using the PSNR ratio metrics versus bitrates of the codecs between MPEG-2, H.263, MPEG-4, and H.264. The result of the comparison shows the outperformance of H.264 with the other codecs.

If you want to deliver the best quality video, we should go with the H.265, but H.265 is still less common than its predecessor, H.264, in the industry, but here we are not trying to prove which has higher quality and all we need to test whether our case is possible or not. So for that purpose, we choose the most used codec, which is H.264.

2.2.4 Papers related to H.264

These are some of the papers that are related to H.264 and some of them are referred in the above section. As even that papers have a clear explanation of the concepts of H.264. So both this section (2.2.3, 2.2.4) papers are taken for H.264 video codec.

The paper [15] explains the whole process of motion estimation and its drawback in it. After that, this paper explains a new technique for motion estimation. And in this paper, they have done a performance evaluation on the methods and results to know the most efficient way for motion estimation. For this evaluation, metrics of motion vector which is acquired from motion estimation and some direct methods like SAD, MAD, and SSD. SAD is used for the motion estimation process in H.264 video codec.

Here in this paper [19], the authors use H.264 video coding standards. They proposed multiple security levels in a net video encryption scheme that meets the requirement of net VOD, video chat, and video conferencing. And by performing some performance analysis on each level of the encryption algorithm.

The paper[20] explains the overview and the history of the H.264 video codec and also explains the VCL design, which is based on motion compensation by the author. The author also clearly explains the advantages of using H.264 and H.264 have coding efficiency and flexibility. This paper[21]explains the fringe benefits of the H.264 video codec. The author clearly focuses on H.264 and the advantages of H.264 are explained in detail.

3.1 Streaming process

Streaming is the act of listening or watching to online media from a remote computer, tablet, or phone. Streaming services record and distribute digital media over the Internet by sending data in small packets over the Internet. Streaming services can also provide other forms of more traditional content ("on-demand" programming) like movies, music videos, television series episodes, and live news broadcasts.

When the video is streamed over the internet, it follows a process to satisfy the viewer. This can be seen from the below Fig 3.1. The streaming process starts to have any video, either live or stored video. This video is sent out for the encoding process to start with the streaming process. In encoding, the video is converted into a digital format that can be compatible with all devices. After the raw video file conversion to digital format, the digital format needs to be stored in a place where all information of the format can't be lost. This process of storing the information is the second step of streaming. The place where we can save the digital format of a video is called a container. Then after storage, we need to deliver the stored video content over the internet. For that, a protocol helps to deliver the data from source to destination. After the video is received by the viewer, the video file needs to be decoded and played to the viewer. These are the steps to be followed when streaming.

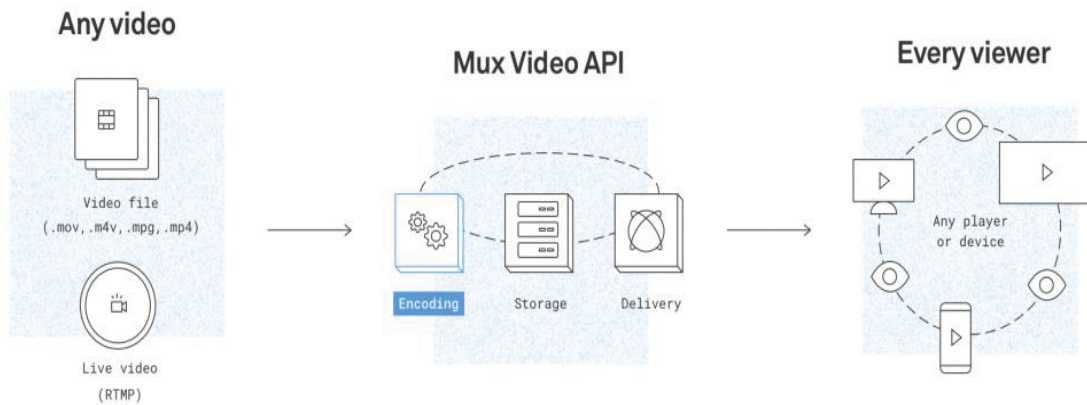


Fig 3.1: A clear view of streaming process[25]

3.2 Encoding

Encoding is the process of converting the data into another format. The format is a representation that can be done in any method without losing the original data meaning. So, here compression is a process of encoding. Usually, the codec uses a special compression format that is paired with a defined encoding. Codec is compressor and decompressor. It consists of two parts named encoding and decoding. Encoding is a process of converting the

video into smaller bytes, which means it compresses the video file size without losing the quality of the file. Decoding is the process of retrieving the compressed video file into its original form. Encoding is done at the start before sending the video for delivery, while decoding is done after the video is delivered to the viewer. Only after the decoding process, the viewer can see the video. Here we use a codec to save the file space and to reduce the usage of bandwidth, as sending the video in smaller bytes, the system doesn't need to use a higher bandwidth.

When coming to the types of codecs, there are some codecs introduced[30]. From the time being, they are updated as per the system configurations and the file size. So, these video compression standards are enhanced by the ISO (International standard organization) and ITU-T (International Telecommunication Union) organizations. Both organizations worked in different ways to publish the codecs. ISO used VCEG (Video Coding Experts Group) to publish the H.26x line standards, while ITU-T used MPEG (Motion Picture Experts Group) to publish the MPEG series. From the below Fig 3.2, the organizations and their published standards with the specific year until today are listed.

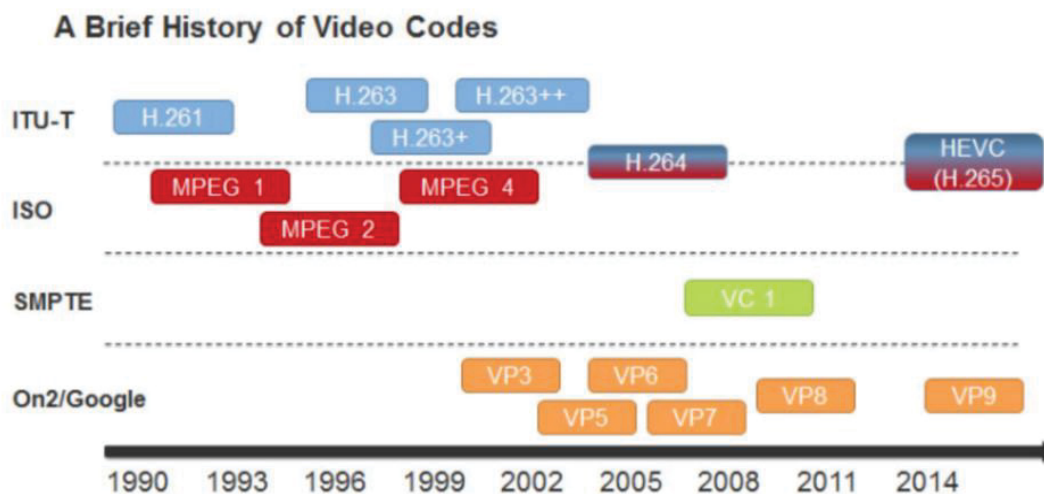


Fig 3.2: Video codecs with their organizations and years [24]

- H.261 - It is one of the first video codecs. It is a standard and is also known as "Video Codec One" or V1. This codec was developed by the team of CCITT (Commission Consultative Internationale Telegraphique et Telephonique) in the late 1980s. This codec actually uses block-based motion compensation to compress video data.
- H.262, MPEG 2 - It was further enhanced by CCITT. When it became H.262, it was meant to be a low cost alternative to H.261. MPEG 2 was developed in the early 1990s. However, as of now, MPEG2 is not used much today.
- H.263 - After the H.261 and MPEG 2's stop by ITU-T, the next codec came into being, which is known as H.263. It was published by ITU-T in 1994, and it is also known as "Video Codec Two" or V2 in short form.

- H.264 - It was created as the successor of H.263 on December 3, 2003 by ITU-T.
- MPEG 4 - It was published by ISO in 2001, and it is also known as part 10 of the MPEG series. It has a wide bandwidth with good quality.
- MPEG 4 has two parts: Part 2 of MPEG 4 standard (ISO/IEC 14496-2) is the standard for audio compression, which is known as Advanced Audio Coding (AAC).
- In many situations, VP3, VP5, and so on. These are used to make low-cost video conferencing systems. These video codecs are used as a video compression standard in VoIP/ Video over IP applications and also in DVD, Wireless, and Satellite broadcasting applications. Mpeg-1 is the predecessor of MPEG-2.

3.3 Storage

After the video file is encoded, it needs to be delivered to the viewer over the internet. For that, the encoded file and the related codec or any other information need to be stored at a place to deliver the contents. So here we need a storage place called a container. A container is nothing but a box that stores the codec, video stream, audio stream, metadata can be seen in the below Fig 3.3.

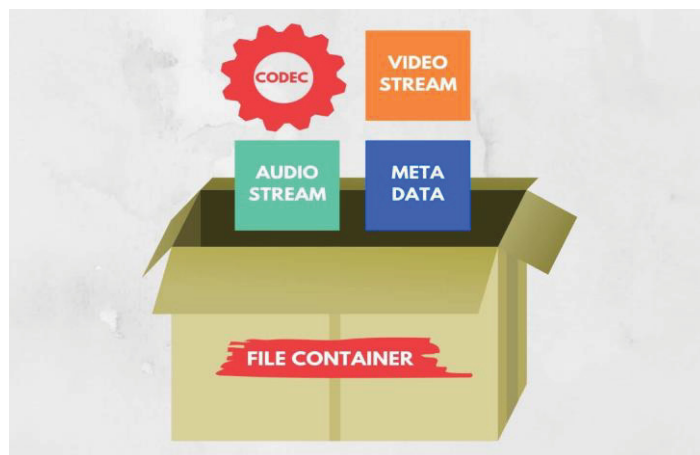


Fig 3.3: Information saved in File containers[23]

The image shows that the container holds all the information and files. As there are lots of devices with different features and support the device requirements, there are a few types of containers introduced. Some of the video container formats are MP4, AVI, FLV, MKV, WMV, WebM, and so on[9]. Here every container has its properties that are different from each other. The properties are divided based on the transportation medium and codec support. And these container formats are compatible only with specific devices. Few are compatible with almost all.

- MP4 - If you have an iPhone or Android phone, the video will likely be saved in this format by default. It can display correctly on other mobile devices, tablets, and PCs.
- AVI - This is the standard for HD video and is used primarily for Web streaming. It

does not have the best compression techniques (i.e., it compresses video more than others)[32].

- FLV - This is a relatively new format that has better compression than AVI. However, it still only supports up to 1080p resolution.
- MKV - This is a high-quality format that supports multiple audio tracks, subtitles, and HD video. If you convert a DVD to computer files, it will likely be in this format.
- WMV - This is a full-featured video format that is supported by Windows Media Player. It has good compression and audio support[33].
- WebM - This is currently the standard for Web streaming and was developed by Google as an alternative to Flash video (FLV). It offers good compression and can save 1080p HD video.

3.4 Delivery

Delivering the content of the video that is stored in the container from one end to the other is done by a protocol. A protocol is a set of rules that control the data traveling from source to destination. Fig 3.4 shows that the protocol takes the delivery part after the encoding and packing of the video. Here from the image, stream data refers to the video, the compression is a part of encoding, the packaging is done in a container, and the protocol takes the delivery part to deliver the contents over the internet.



Fig 3.4: Process of video streaming [7]

Streaming protocols are used to deliver different types of media over the internet. The streaming protocols can provide fast video delivery by using separate streaming servers[31]. The most common streaming protocols are RTMP, RTSP, HLS, MPEG-DASH, and so on.

- RTMP - It is a TCP based protocol. It is also called Adobe's RTMP. This is now an open specification[11].
- RTSP - It was developed by Apple Inc. It is a network control protocol that can be used to connect client and server components of streaming video applications. RTSP is included in most streaming software such as QuickTime, Adobe Flash, and Real Media.

- HLS - HTTP Live Streaming enables an adaptive streaming format for delivering a single media resource over multiple bandwidths and has more functionality than RTMP.
- MPEG-DASH - MPEG DASH and DASH Express standards are based on DOCSIS 3.1 specifications. MPEG-DASH is designed to be a streaming media delivery scheme that can scale across today's broadband networks and the future of ultra high-speed Internet.

Up to now, the streaming process is explained so that the steps that are done for the thesis can be clearer. Here the main idea is to use obscuring method into a video and make it look like noise and transmit it over the internet to make it more secure, and even if some breach occurs, the video can't be retrieved back without knowing the encryption process. The obscuring method is to hide the information, and this information in the video is done by the existing algorithm called the obscuring algorithm. This algorithm makes the video looks like a noise from the information within the video.

Now as to start the experimentation, the first step is to encode the video. So to encode the video, a codec needs to be used. For that, H.264 codec is used as it is compatible with most of the devices.

3.5 H.264 Overview & working

H.264/AVC is the most widely used video codec when compared to other codecs. H.264 is the result of the joint effort of the organizations' named ISO (International Standards Organization) works on MPEG and ITU-T (International Telecommunications Union) works on VCEG. This is the reason that H.264 is also known as MPEG-4 PART 10/AVC. H.264 video coding standard was developed over four years of time span. H.264 is very flexible and is compatible with a wide range of applications with very low and as well as very high bitrate requirements. Some features of H.264 are image enhancement filter, high-definition resolution, interlaced/progressive scan mode, and much more.

The H.264 has a long chain of components from source to destination to pass through. This chain of components is divided into two parts they are encoding and decoding. The encoding contains color image processing, motion estimation, motion compensation, intraframe prediction, and image compression model. The image compression model has three components named DCT, quantization, and entropy encoder. In the same way, the decoding contains the entropy decoder, de-quantization, inverse DCT in the image decompression model, and the rest are intraframe prediction, motion compensation, and motion estimation. The decoding process follows the same as the encoding, but it's the reverse process.

Here the experimentation explains the steps this thesis goes through. Firstly, this section explains the encoding and decoding part of the encrypted video. The encryption is done by the obscuring process.

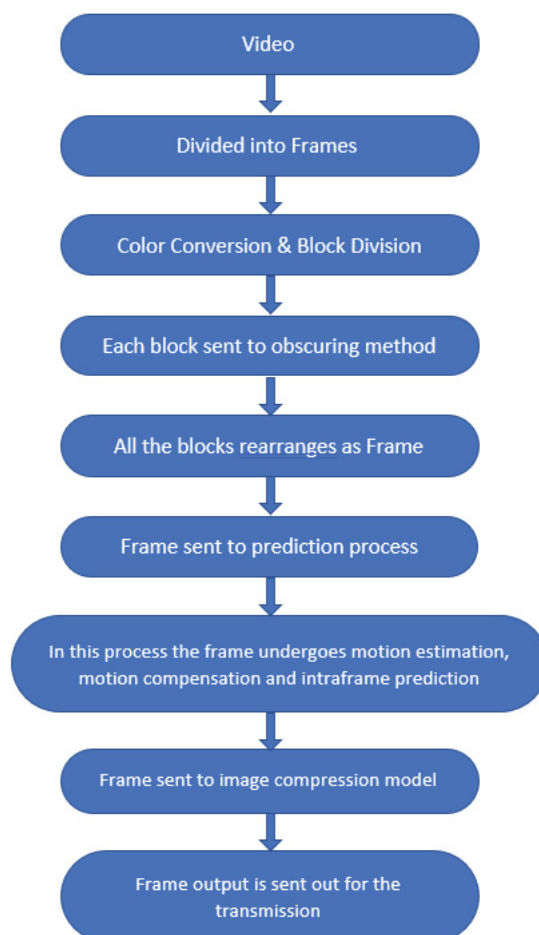


Fig 4.1: Flowchart of encoding process

Here the Fig 4.1 is a flowchart for the process, takes the video as input and gives an overview of encoding process. Here the encoding process is divided into two parts and those two parts are included in the flowchart.

4.1 Encoding process

The encoding process starts to take a video and the video is divided into frames. A video is a periodical sequence of images called frames. Each frame is a single image, and that frame can be divided into blocks. This can be seen clearly through the below image and then each macroblock has its color components RGB or YCbCr.

Then the pixel information is shown in one of the macroblocks. This is a small explanation of what is going to be done in the further process. This Fig 4.2 can give a small clarification. As the video is a sequence of images, from these a frame is shown and the block division of it is also shown. And each macroblock contains the pixel values is also shown in the Fig 4.2.

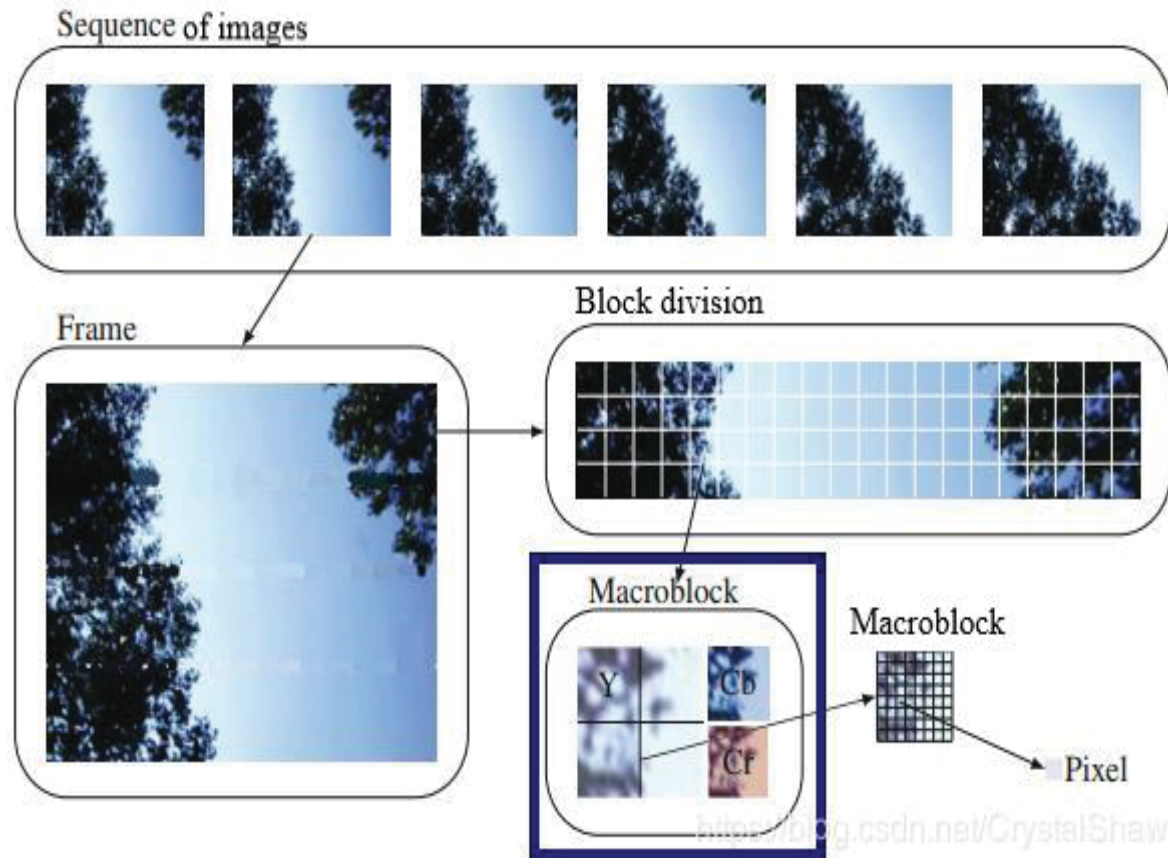


Fig 4.2: Explanation of frames and their blocks[34]

From here, the encoding process is divided into two steps. In the first step, the frames of the video are encrypted and in the second step, the frame is under a pre-processing level and then the compression will complete the second step of the encoding process.

4.1.1 First part of encoding process

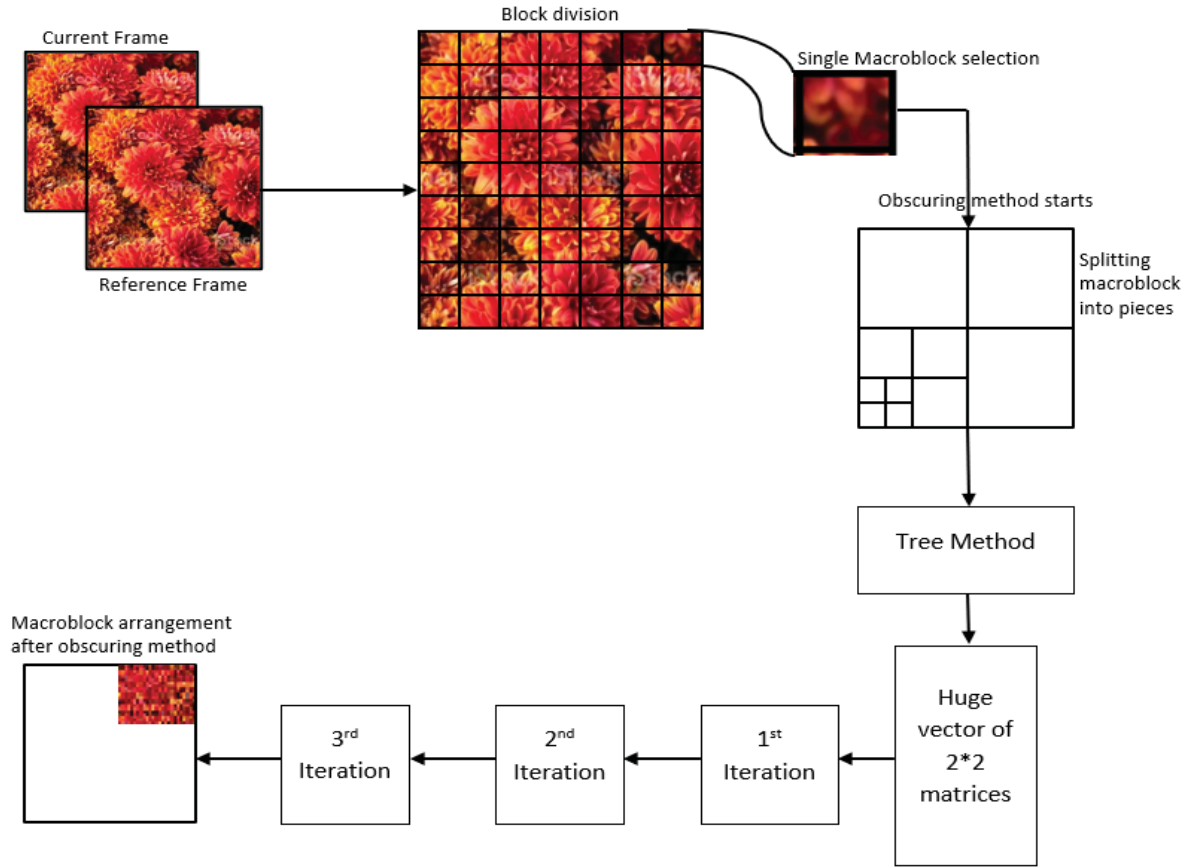


Fig 4.3: First step in encoding process

The process starts by having the first two subsequent frames of the video file can be seen in Fig 4.3. The first frame is the reference frame(I frame), and the second frame is the current frame(P frame). These two frames are sent into the color image processing.

4.1.2 Color image processing

Each frame has a 2D array of pixels. The pixel of the frame consists of R, G, and B color components. RGB (red, green, blue) color model is a widely used model. In color image processing, the pixels are converted from the RGB model to the YCbCr component as the human eye will not differentiate the difference in frame boundary. As shown in the below Fig 4.4, the conversion of RGB to YCbCr components is shown clearly in each component.

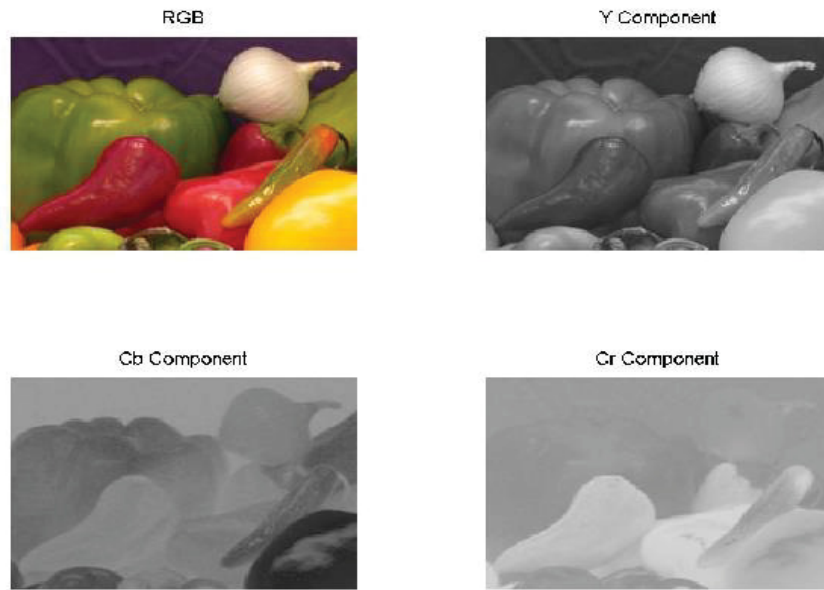


Fig 4.4: Color image processing from RGB to YCbCr[35]

Y is the luma color component, while Cb and Cr are the chroma components. After converting both the frames into the YCbCr color component, which looks like a greyscale frame, then division of the frame into macroblocks is done.

4.1.3 Obscuring method

Here starts the obscuring method that hides the information in the video into noise. Then first, the reference frame is selected for the block division into 32-bit size blocks. The block division of a frame is shown in the Fig 4.5, and the macroblock contains the frame information.

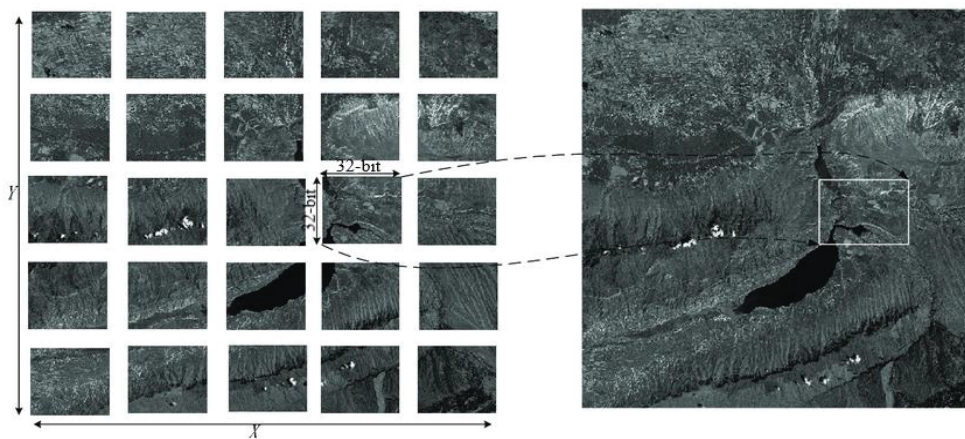


Fig 4.5: Frame division into blocks of 32-bit size[52]

Each macroblock is selected and divided into smaller pieces by a process. The process is the block is first divided into four equal parts. Each part is again taken as the input and again divided into four equal parts. Now there are 16 parts, and this process of dividing takes place until the output is a 2×2 matrix can be seen from the Fig 4.6. This process of splitting the block again and again in a function is called recursion. This recursion process ends until a stop function which is predefined by the correlation coefficient. So the output is of the block of different numbers of 2×2 matrices and a huge vector. The process of dividing into matrices until a 2×2 matrix is called a tree method can be seen from the Fig 4.7. As the splitting of the block is done, then immediately, the divided parts are numbered accordingly, and this numbering is called indexing.

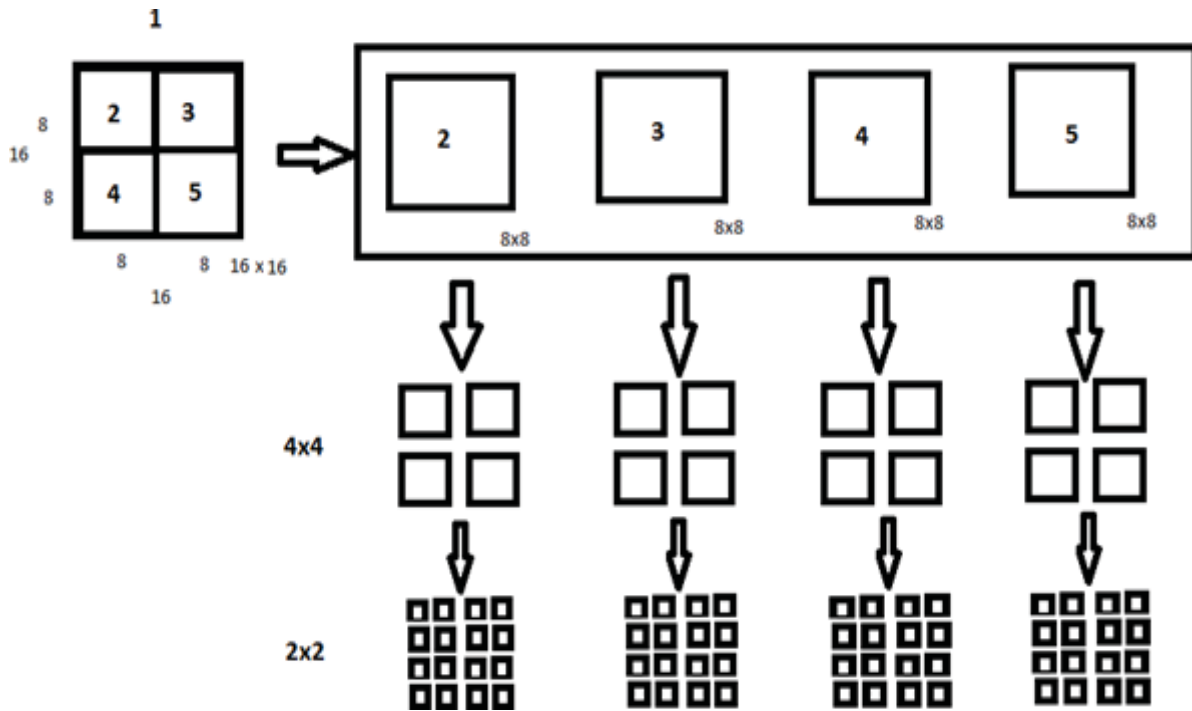


Fig 4.6: Splitting of block into pieces[37]

Then the huge vector of 2×2 matrices is arranged randomly into the block and sent to the first iteration. These iterations are to make the information inside the block difficult to understand by any other person. The iteration is to mix all the 2×2 matrices in the block. This way, there are three iterations that totally make the block look exactly like a noise. Then this block is stored in a new empty frame so that the same process is repeated to all the other blocks in the frame and rearranged back to the same exact places as the original frame.

As the frame is divided into 32-bit size blocks, all the blocks should be sent under the same process. Then, after all the blocks of the frames undergoes this process, in the same way, this must be done to the other frame, which is the current frame. Then the output of these frames looks like noise. Then the frame is divided into macroblocks for prediction techniques.

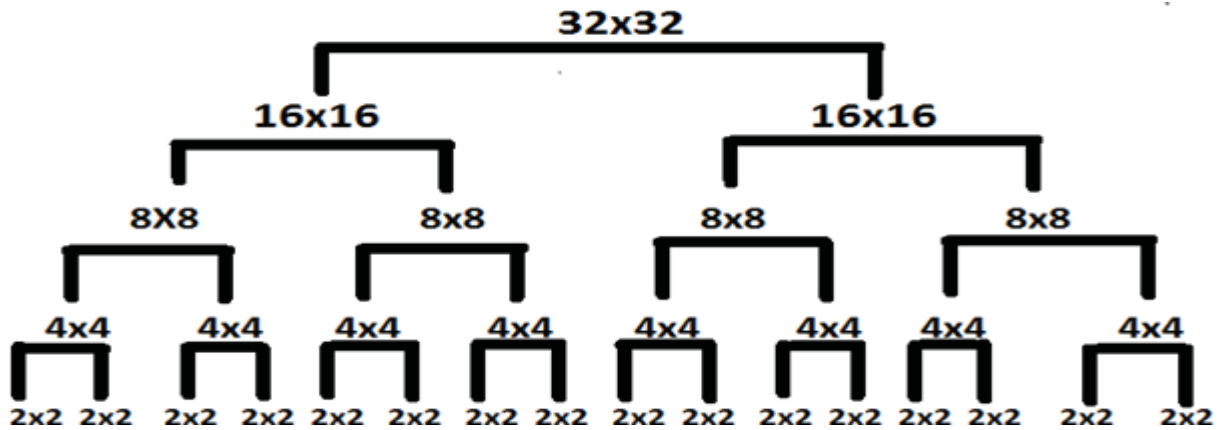


Fig 4.7: Block pieces arranged into Tree encryption method[37]

Here the first part of the encoding process ends, the output from this part is the reference frame and the current frame looks like noise. Then these two frames are sent to the second part of the encoding process that does the prediction techniques and the compression model. This can be seen in stepwise format from the below Fig 4.8.

4.2 Second part of encoding process

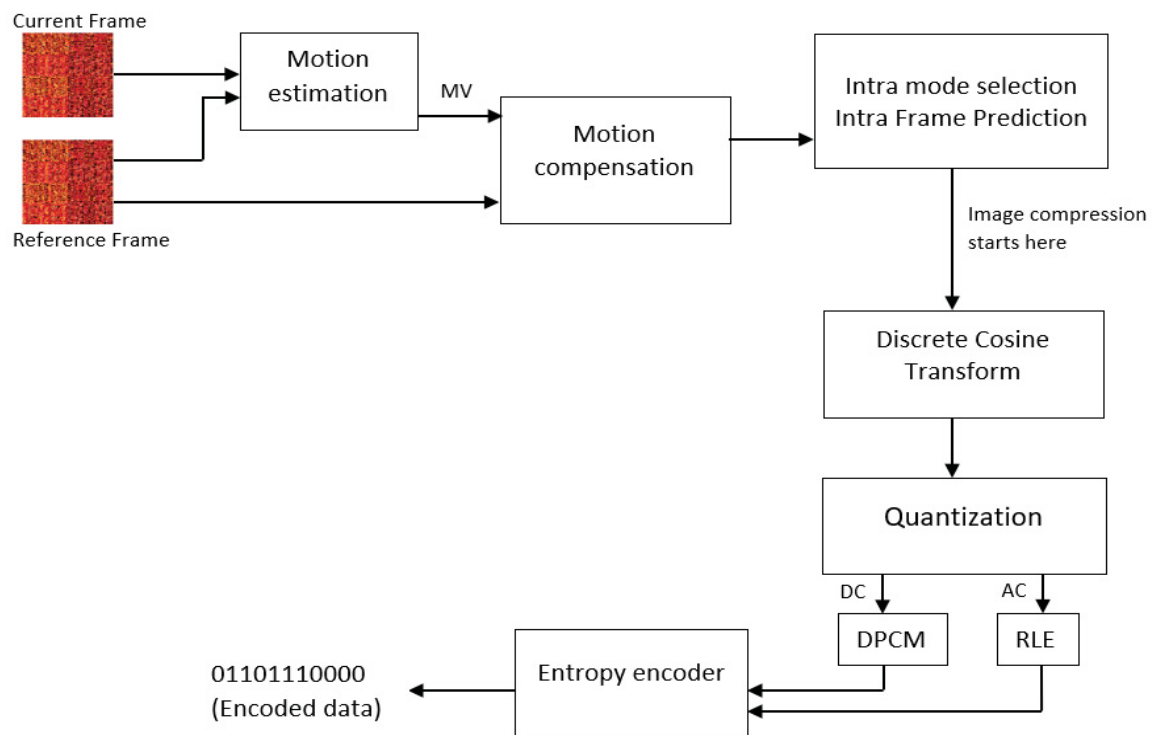


Fig 4.8: Second step in encoding process

4.2.1 Motion Estimation

The output reference frame and current frame look like noise are given as input to the motion estimation. Inter frame prediction is the other name for motion estimation. Both the frames are divided into 16×16 macroblocks before processing. The motion estimation removes the

temporal redundancy by searching the position of the current macroblock of the current frame with the reference frame. Here redundant data refers to the same data between the two frames. This prediction of macroblocks between two frames can be seen in the below Fig 4.9. It shows the prediction of temporal redundancy from two frames and each block is of size 16×16 MB is the macroblock, and the search area is the selected part in that selected part the search will be done.

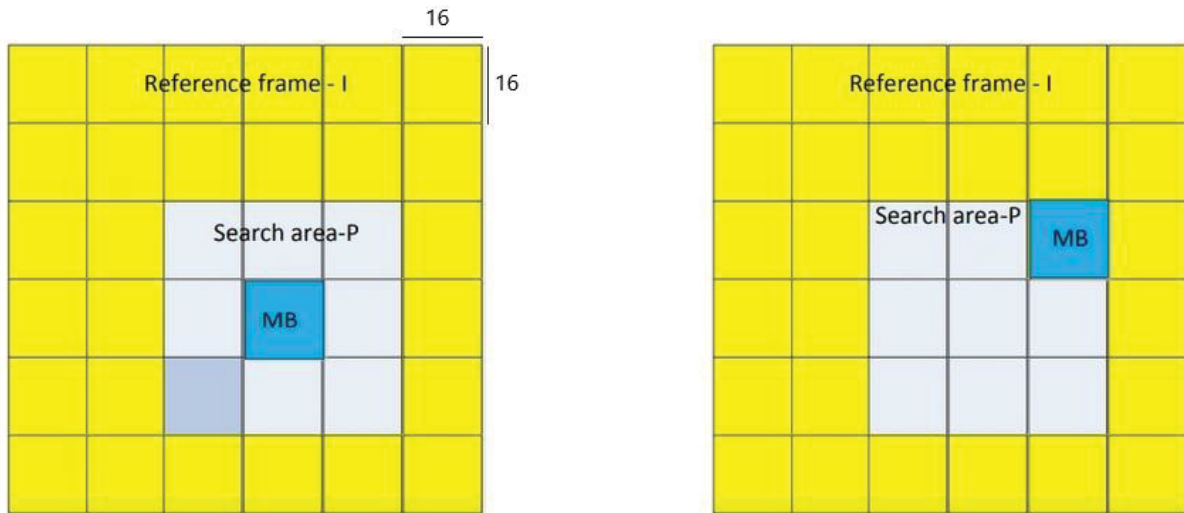


Fig 4.9: Comparison of Macroblock in search area between 2 frames

Suppose any of the macroblocks matches, the x-y coordinates of that macroblock are stored in the motion vector. There are a few algorithms that are used to search for the match in the frames. An exhaustive search algorithm, three-step search algorithm, four-step search algorithm, and novel four-step search algorithm are the search algorithms. By taking high computational cost, uniform search pattern, image quality as the drawbacks of the algorithms, the Novel four-step search algorithm is considered the best of all algorithms.

4.2.1.1 Novel four-step search algorithm

When the current frame is predicted with respect to the past and future frames, then that frame is called a B frame or bidirectional frame. This search algorithm is the enhanced version of the previous algorithms and also has less number of search points. Here a cost function is used that is based on the distance between the two macroblocks. The search algorithm starts at 9 points in a 5×5 search window with a step size of 2 in the center. So if the minimum cost is located at the center, the search window jumps to the 3×3 search window with a step size of 1, or else it continues to the next step search. The next step is to maintain the search window to 5×5 , so if the minimum cost is located at the other five search points, then go to the next step. Here after that, the same process is repeated, that continues to have the search window to 5 points. And then, in the last step it reduces the search window to 3×3 and the step size to 1 can be seen from Fig 4.10. The Novel four-step search algorithm reduces the search points faster than when compared to other algorithms[22].

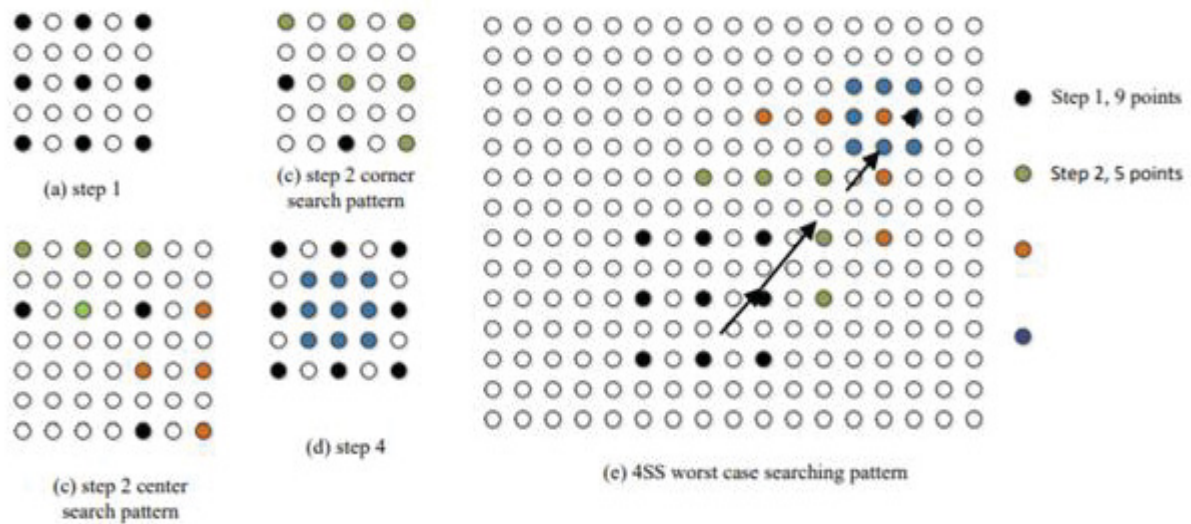


Fig 4.10: 4ss step search algorithm

The data related to the diagram was obtained from the mentioned references[53][54][55].

By using the Novel four-step search algorithm, the search of the macroblock is done. So here the SAD, MAD, and MSE algorithms decide whether the match between the macroblock of the current frame with the reference frame is the best match or not. After the best match is found the x, y coordinates of the matched block are saved into the motion vector.

4.2.2 Motion Compensation

The motion vector and the reference frame are taken as the input to the motion compensation process. In motion vector, the removal of temporal redundancy is not done perfectly. So in the motion compensation, the temporal redundancy is removed with respect to the motion. Here the motion detection is done either by fixed block size, variable block size, overlapped block size, half-pixel, or quarter pixel. By using any of these techniques, motion detection can be removed if a blank frame is taken and filled with the information in the motion vector and macroblocks of the reference frame. The output of the motion compensation process is the compensated frame. Here motion compensation regenerates a frame that is more like the current frame, as the regenerated frame is from the motion vector and macro block of the reference frame. The regenerated frame is also called a motion compensation frame. Here the motion compensated frame can be seen in the below Fig 4.11. The first frame in the figure is a general grey scale image and the second frame is its motion compensated frame. But here, as the motion compensated frame is done to the noise frame, there cannot be seen any difference because the human eye cannot detect the change between two noise frames.



Fig 4.11: Reference frame and motion compensated frame[38]

The residual frame is the difference between the current frame and compensated frame as motion compensation can be either implemented by one of the block size methods. And all the other block size methods have some drawbacks like using more bits bigger dimensions. The variable block motion compensation is used to implement the motion compensation as it requires less no of bits by merging the macroblocks into the larger blocks, which have less motion.

4.2.2.1 Variable block motion compensation(VBMC)

The VBMC uses a method of merging the normal sized macroblock into the larger block that saves lots of storage space. This changing of a normal sized block to a larger block is based on the desired threshold value. Here the idea is to merge the still background into the larger block and if any small motion is detected, this is divided into the smaller blocks so as to capture the motion in the macroblock clearly. The raster scan order is used in the encoding of VBMC. The fixed block based division is also called FBMC, and this can be shown in the first image in Fig 4.12(a). As the size of the blocks is fixed, the small motion can't be detected, so here it uses the VBMC (Variable Block motion compensation). This can be seen from the second image in Fig 4.12(b) that clearly detects the image and the storage space is saved a lot when compared with FBMC.

There is a root macroblock and the next macroblock is used to evaluate the difference between them and then the decision of merging needs to be done. In the same way, the root macroblock is differentiated from the other macroblock. If that is a match, then it moves to the next position or else it moves to the next row.



Fig 4.12(a) FBMC



Fig 4.12(b) VBMC

Fig 4.12: FBMC and VBMC[36]

The decision is based on the difference between the macroblock's threshold values. If it is less than the adjustable threshold, then the two macroblocks can be merged or else, it's a mismatch. If there is a mismatch, then the adjustable threshold values are higher, so the raster scan order needs to proceed to the next row, and the same process continues until the mismatch in the upper row. Then this process continues until it gets a mismatch in the same column where the root macroblock is located. The next step is to merge the sections of the matched macroblocks of the frame. The merging section can be clearly seen in the Fig 4.13.

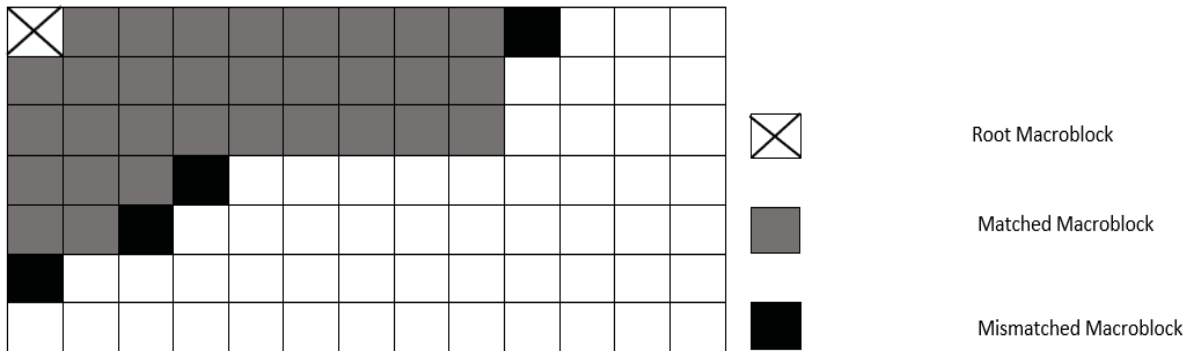


Fig 4.13: Matching of macroblock with the Root macroblock

Here the first mismatch of the 1st three rows can be taken as the wide area to merge and then the mismatch of the column is before the mismatch of the last row is taken as the height to merge the macroblock. Now, this type of division results in 2 candidate blocks. These two candidate regions are different in size and shape. The candidate region with a higher number of blocks is selected to merge then all the other region macroblocks are recorded for the

decoding process. This process continues until the end. Here there can be merged or unmerged data after completing the process. Here in motion compensation, the macroblock size can be 16×16 or 8×8 , or 4×4 according to the detection motion in the frames the division can be in such a way that can be shown from Fig 4.14.

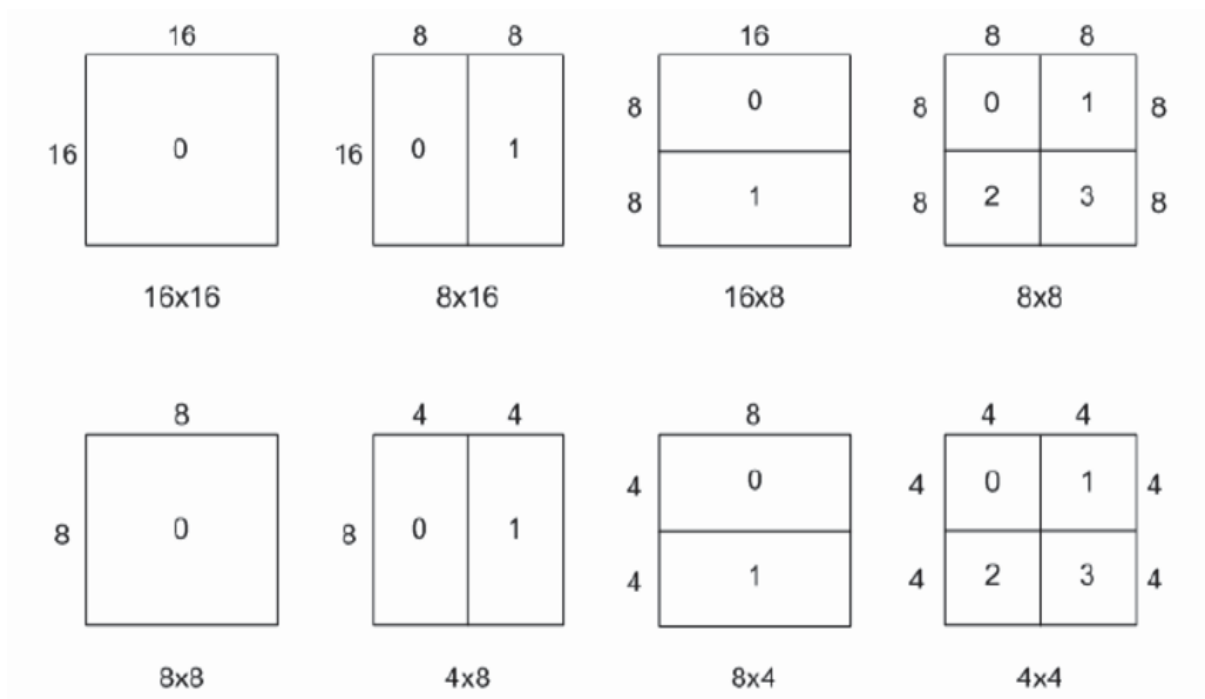


Fig 4.14: Macroblock sizes w.r.t motion[39]

4.2.3 Intra frame prediction

This is the compression that does on the information of individual frame and it limits the information with minimum loss. The spatial redundancy is removed from the frame in intra frame prediction. Spatial redundancy exploits the redundant data within a frame. It removes the redundant data so that it has a very small amount of information. Spatial redundancy consumes less processing time as it particularly needs to look at a frame at a time. The macroblock division in the motion compensation is based on the computation cost. As the number of blocks increases, the number of vectors to encode them increases. This can be seen from the Fig 4.14. Here for this, an 8×8 macro-block is used for implementation. It has nine different modes. These modes are used on the previously coded macroblock that results in intra coded macroblock.

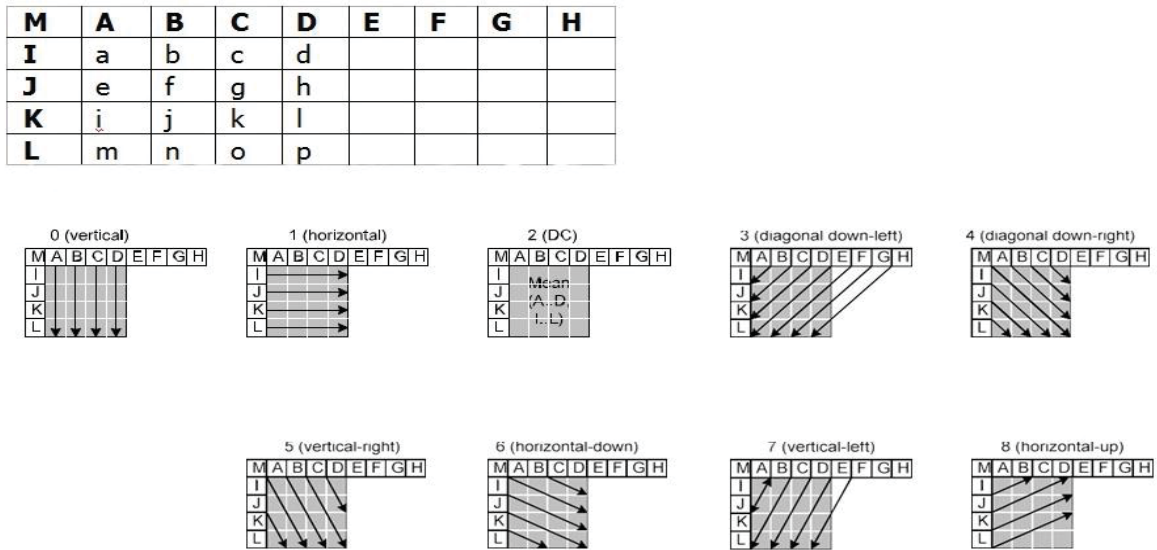


Fig 4.15: Intra mode predictions of all directions [40]

This intra prediction has 9-intra prediction modes can be seen in Fig 4.15. Each has its direction and particular method of use. So for 8×8 macroblock, intra modes of mode-0, mode-1 and mode-2 are used for implementation. In the 8×8 macroblock, each block is sent into the mode selection. From the modes, one of the modes is selected and the selection is done by the SAE, or MAD, SAD. So the input for this is the reconstructed macroblock. Here the predicted macro block is subtracted from the original macroblock and the output is a residual macroblock. As all the residual macroblocks form into a residual frame.

4.2.4 Image compression model

The residual frame is the input to the image compression model. The image compression model consists of the DCT, Quantization and Entropy coding. Here after the encoding process, the compression needs to be done so as to compress the information. Here as the residual frame is of 8×8 macroblock size. Each block goes to the DCT and then Quantization and Entropy coding. As from the below Fig 4.16, the frame is given then after the whole process, and the image output is some codeword. This image is just an example to show what the output can be. In our case, the image will be the noise image after the intra frame prediction the residual noise frame will be sent through all the processes of the image compression model.

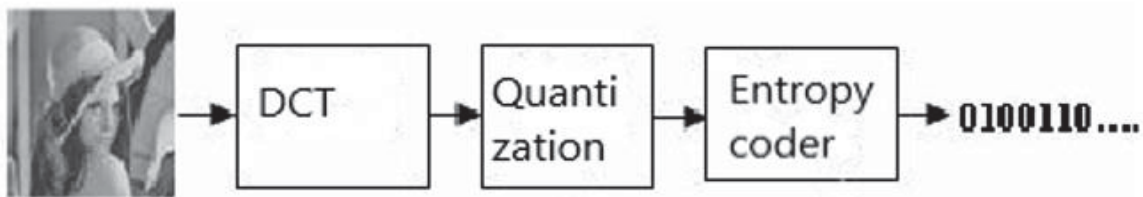


Fig 4.16: Frame going through the image compression model steps[41]

4.2.4.1 DCT (Discrete Cosine Transform)

Here each block of the residual frame takes an input and converts the signal into the frequency domain matrix. So in the residual frame, the DCT is applied to each and every block of the frame. This DCT works on real numbers and it uses a 1D DCT for each and every row and column.

4.2.4.2 Quantization

Then the 8×8 macroblock size is ready for quantization. Quantization is the process of converting the values to small values. The quantization principle is working in a way that a low frequency signal is more sensitive than a high frequency signal to HVS. So quantization does low compression to low frequency signal and high compression to a high frequency signal. All the values in DCT and Quantization are in the matrix form and the values are the totally real number. This matrix form of numbers is the values of the macroblocks in a frame. So after quantization, the matrix is sent to the entropy encoder.

4.2.4.3 Entropy Encoder

Here the matrix which is obtained after the quantization is converted using different types of methods. Here, two of the methods can be seen. The alternative scan and the zigzag methods are the two methods used in the entropy encoder to change the frame into bit stream.

Before the entropy encoder, there are two steps that separate the AC and DC component. The DPCM (Differential pulse code modulation) and RLE (Run length coder) are taken as a part of the entropy encoder. Here the DPCM takes the DC components that have the higher values. The entropy encoder is the last step of image compression. The exact process is after the AC and DC component is divided, there comes the serial stream of the ZIGZAG process. In the second step, these streams are converted into the symbols by run length coder and then a code word is added to the symbols that are generated according to the values from lower to higher by Huffman coding. All the image compression is done, and the output is sent for transmission.

4.3 Decoding part

In the decoding part, the same exact process is repeated but in the reverse order as the encoding process is of two steps the same repeats in the decoding process. Here the second step comes at the beginning of the decoding part in the reverse process. After the output is sent out for transmission and received at the decoder end, the first thing to do is an image decompression. The output is sent to the entropy decoder, de-quantization, and inverse DCT in the image decompression. The entropy decoder starts to convert the codewords into symbols in the same process as the entropy encoder. Then these symbols are changed into the serial stream. This stream is converted into the matrix, and the matrix is sent into the de-quantization as an input.

In de-quantization, the higher and lower frequency components are removed from the matrix. Then after the removal, the matrix is sent to the inverse DCT, and the matrix is again converted to the residual matrix.

Here in the quantization or de-quantization, the reconstructed signal is different from the original signal, and this shows that some of the information is lost during the process. If the process continues to inverse DCT the error might increase. So, to enhance the image quality and reduce the error, H.264 introduces a deblocking filter in the decoding process. As the change in frequency components results in an error in the block boundaries that can be identified by the human visual system. This can be known as a blocking artifact. So, the filter used here is to reduce the artifact and to reduce the error by replacing some of the original values with the filtered values so that even if lose them, the filtered values can be replaced by the original values. After this, the residual signal is given to the intraframe prediction, motion compensation, and motion estimation. Here the second step ends and the frame is sent to the first step to remove the noise from the frame.

In these, the decoding process is the same as the encoding one but it does that in the reverse order so as to get the desired output. The output from motion estimation is sent to remove the noise from the frames. So these frames are again divided into 32-bit sizes and each block is sent individually to the iterations and the tree mode to remove the noise from the blocks. And these blocks are rearranged back into the same order of the frame. The tree method for decryption can be seen clearly from the below Fig 4.17.

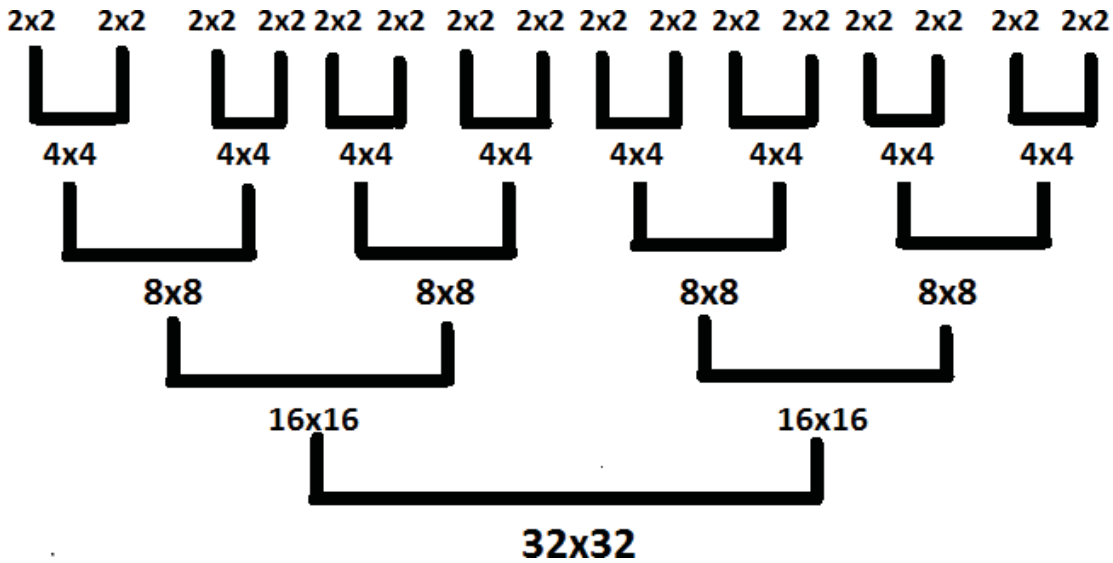


Fig 4.17: Block pieces into block with Tree Decryption method [37]

As seen from the above tree method decryption, the process is reversed from the encoding process to retrieve back the contents. Then after the removal of all the noise then, that's the decoded output, which is shown as the output to the viewer. This way, all the frames go through the same process, through the obscuring method to completely have the video encoded and decoded.

From the encoding and decoding process, the obscuring method is successfully encoded and decoded back. Now, as explained in the background section, the first process of the streaming is to encode and this is done. The second process of streaming is to store the encoded files in the video container. The container used to store the information here is MP4, as it is

compatible with lots of devices. Here the encoded information is stored in the container and then decoded into the video file back normally. Hence the encoding and storage are done before entering into the third process of streaming, i.e., delivery. The process up to now has to be sent for the performance evaluation to see how it works. This performance analysis can be done on video factors by taking different resolution videos with different textures.

For checking the performance, a database of videos and their information needs to be stored to explain them in detail. So, the below section is just to provide the videos and the information that are used in the performance evaluation.

4.4 Materials

If we consider a database consists of some videos from Fig 4.18, 4.19, 4.20, 4.21. Each video has different properties, and these videos are selected based upon the method and the performance of the videos to process them. Each of the video properties is shown in a tabular form Table 4.1. These properties are used just to show the clear details of the videos which are selected.

	video-1	video-2	video-3	video-4
File formats	mp4	mp4	mp4	mp4
Total no of frames	26	20	9	24
Video size	449kb	246kb	22.3kb	199kb
Frame height	352	240	240	240
Frame width	480	320	320	320

Table 4.1 Properties of different videos

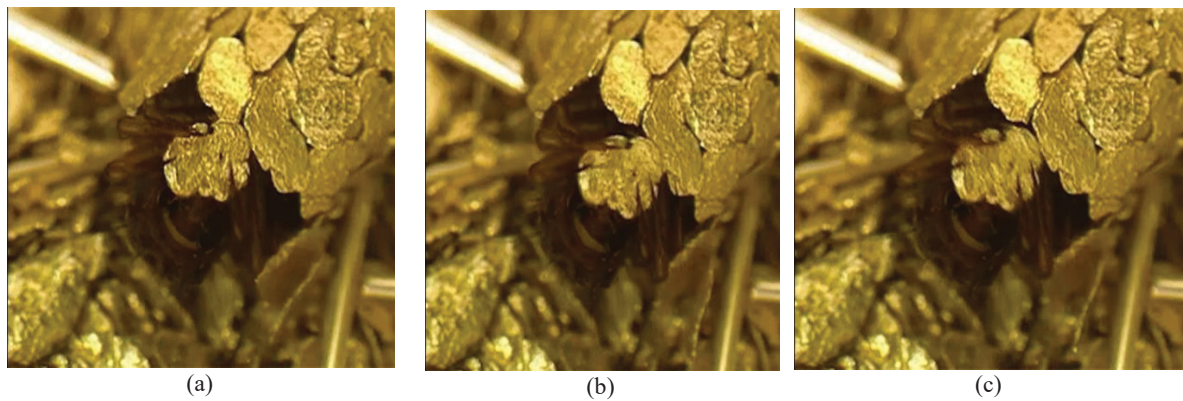


Fig 4.18: First three subsequent frames of video-1

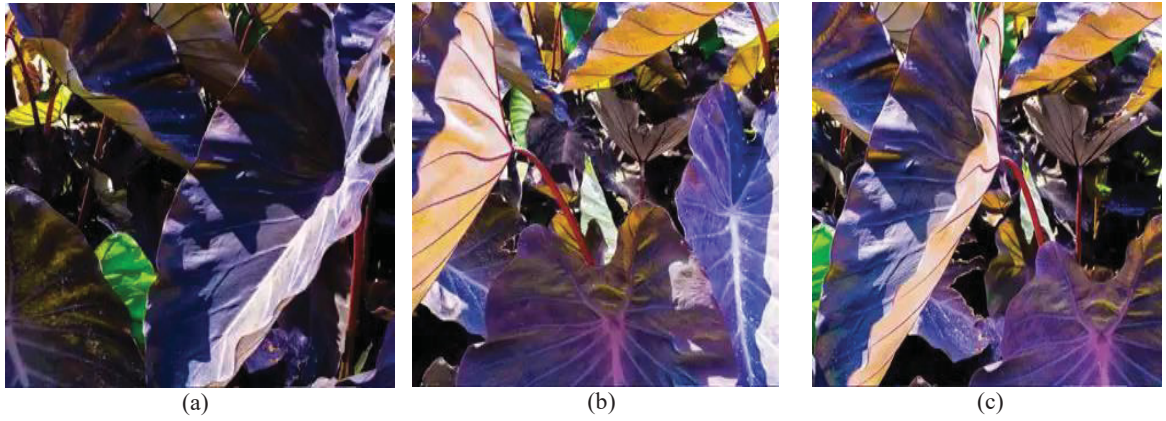


Fig 4.19: First three subsequent frames of video-2

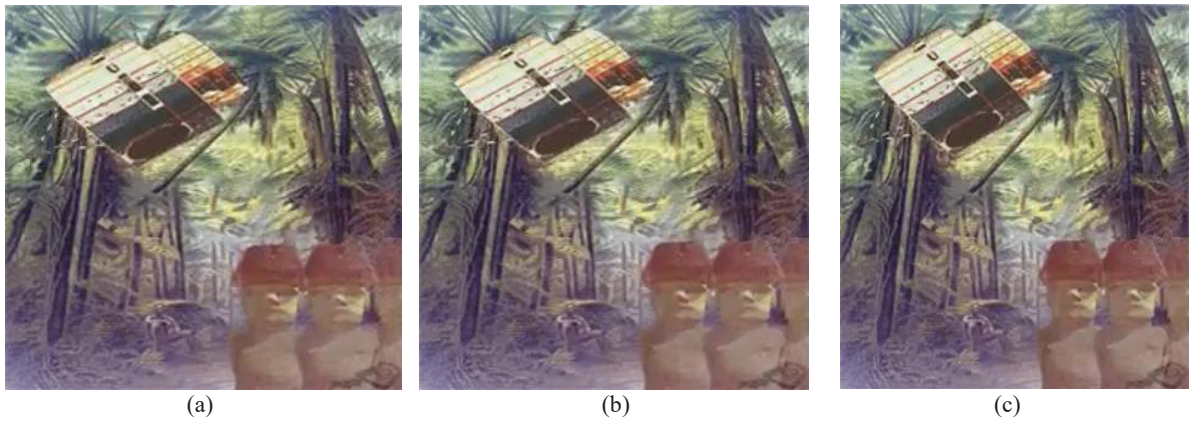


Fig 4.20: First three subsequent frames of video-3

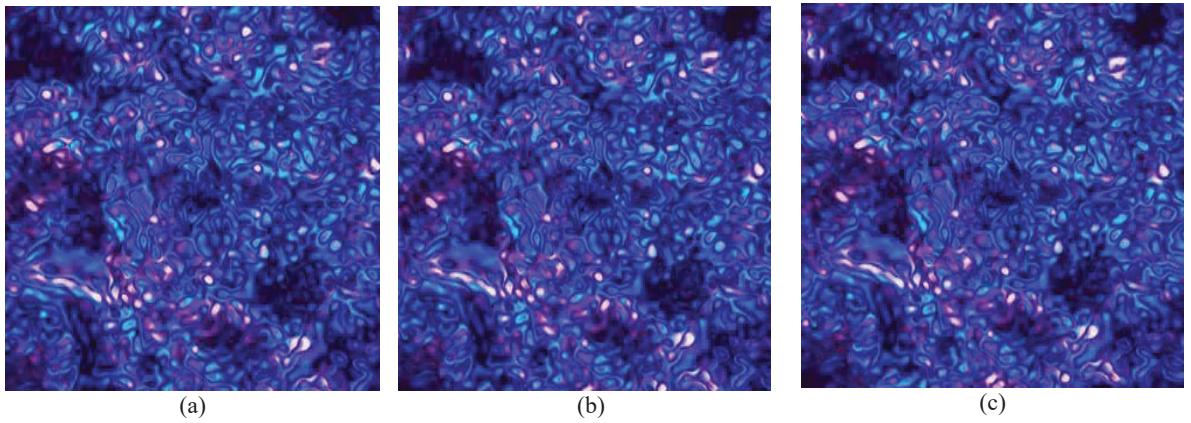


Fig 4.21: First three subsequent frames of video-4

5.1 Encoding and Decoding results

The obscuring method is used to encrypt the video file, and as per the explanation from the above chapter, this chapter has the results after the encoding and decoding process. These results are used in the performance analysis to check the feasibility of the method based on the video factors.

These are the results for video-1 from below Fig 5.1. These results comprise the steps of encoding and decoding explanation in the above **chapter 4**.

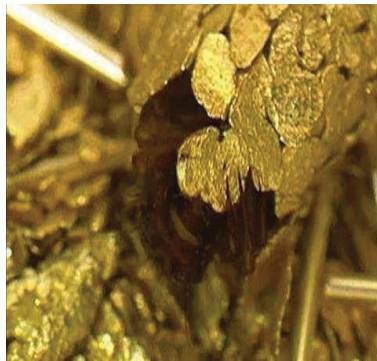


Fig 5.1(a)
Original frame

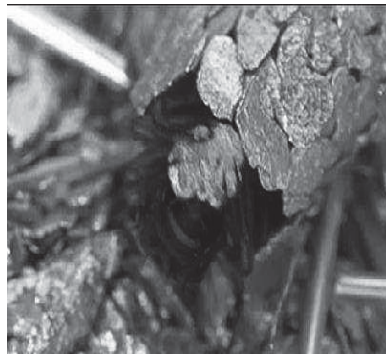


Fig 5.1(b)
Grey scale frame

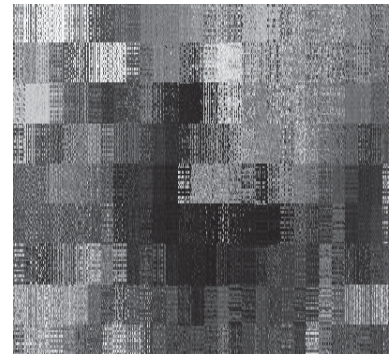


Fig 5.1(c)
Encrypted frame

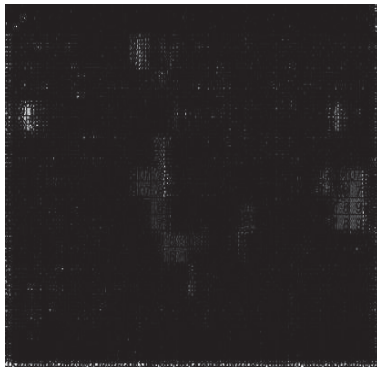


Fig 5.1(d)
Block division frame

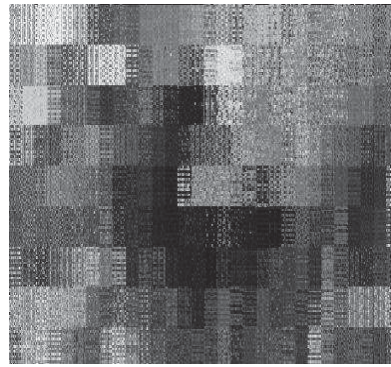


Fig 5.1(e)
Decrypted frame



Fig 5.1(f)
Output frame

Fig 5.1 Results of video-1

These are the results from video-1 that are processed from the method. Here these results are explained at what step they are acquired to have a clear view. As the video is divided into frames in the first step of the method used in the encoding process, Fig 5.1(a) is the first frame of video-1 before color conversion. Fig 5.1(b) is the frame that is generated after the image color processing in the encoding step and can be called a greyscale frame. Then the frame in Fig 5.1(c) is after the obscuring method of the frame, and this frame totally looks like noise. The information in this frame can't be guessed easily, even if it's hacked. So this frame can be called the encrypted frame. Then after the obscuring method, the frame is sent to the following steps, where the block division is done with and without motion. Here the frame cannot be seen clear as there is a lot of motion in the frame, so the block division is done

accordingly. This frame from Fig 5.1(d) can be called a block division frame. Figure Fig 5.1(e) is the frame that is sent to the decoding process after removing all the block divisions from the frame. This figure is named the decrypted frame. And then finally, the figure Fig 5.1(f) is the output frame after going through the all decoding process that is explained. These are the results to show from the video-1 so that the viewer gets the overview at which step the frame is and what exactly the frame is. Then the performance analysis is done for the whole process, so to check the feasibility and quality of the method. For the performance analysis, this thesis uses two video factors.

1. Scaling factor
2. Texture factor

5.2 Scaling factor

The scaling of the video is a change in the size of the video. The size refers to the pixel resolution of the video. The pixel resolution plays an important role in the video. So, this factor is chosen to reduce and increase the pixel resolution of a single video and observe whether this factor affects the video while using the method or not. A single video with different pixel resolutions is taken to generate the results for this step.

The selection of a single video with different pixel resolutions is to see the change in the output and compare them with one another so that the cause of resolution change matter or not can be observed. Outputs are generated for each resolution, as shown below. So, the performance of the scaling factor can be measured by taking the mean error values and variance error values of the input frame (greyscale frame) and output frame. By using the error values of the greyscale frame and the output frame, an error image between them can be generated. By using these values, the difference is seen from the input frame to the output frame and can calculate the statistical values like mean error values and variance error values using an error image.

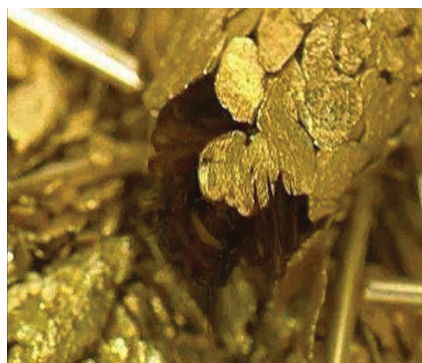


Fig 5.2 (a)
Original frame



Fig 5.2 (b)
Greyscale frame

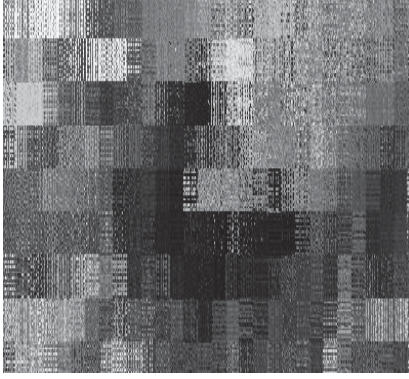


Fig 5.2 (c)
Encrypted frame



Fig 5.2 (d)
Output frame

Here a video of 340×480 pixel resolution is selected for the execution. Then the video is divided into frames. The first frame is the original frame of the video that is selected, as shown in Fig 5.2(a). The first frame of the input changes into the greyscale frame, as shown in Fig 5.2(b). Fig 5.2(c) shows the encrypted frame, and this frame is sent for transmission. The output after all the processes is the output frame shown in Fig 5.2(d). The greyscale frame and the output frame from Fig 5.2(b) and Fig 5.2(c) are used to generate the mean error values and variance error values.



Fig 5.2(e)
Error image

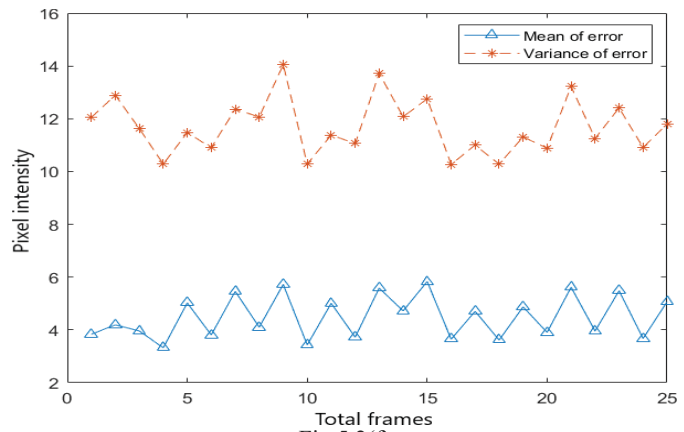


Fig 5.2(f)
Graphical representation of total frames and pixel intensity

Fig 5.2 video-1 with original pixel resolution

Here the error image is calculated for one frame by taking the difference between the greyscale frame and output frame, as shown in Fig 5.2(e). So, by using this error image the mean error values and variance error values are calculated for a frame. Firstly the error frame is the difference between a frame of input and output frame, and from this error frame, the mean error values and variance error values are calculated. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph, as shown in Fig 5.2(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity.

Now, as from the observation, the output for the original pixel resolution video. Here the same video as shown in figure Fig 5.2 is taken, and the pixel resolution of the video is reduced to see how that affects the video pixel factors.

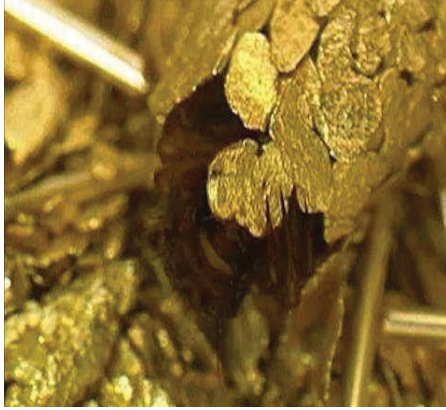


Fig 5.3 (a)
Original frame



Fig 5.3 (b)
Greyscale frame

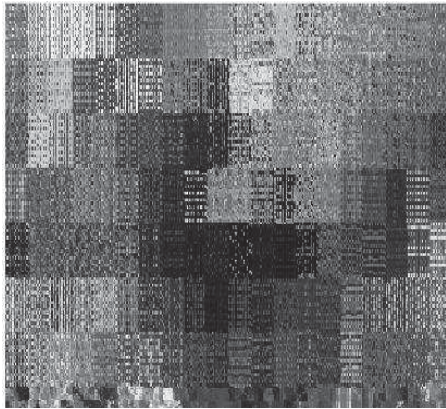


Fig 5.3 (c)
Encrypted frame

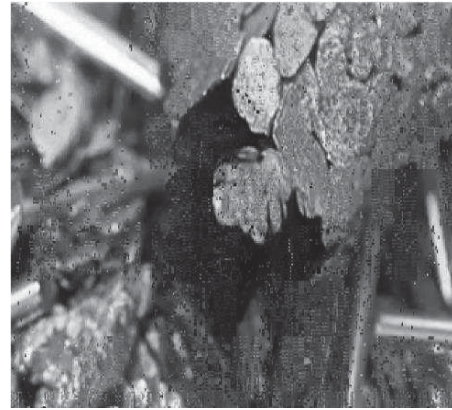


Fig 5.3 (d)
Output frame

Here the pixel resolution of the video used is 240×320 . The original frame of the video is shown in Fig 5.3(a) to have a view of the video after the pixel reduction. From Fig 5.3(b), the greyscale frame, Fig 5.3(c) encrypted frame, and Fig 5.3(d) output frame can be seen. The greyscale frame is the first frame of the video. The greyscale frame and the output frame are used to get the mean error values and variance error values.

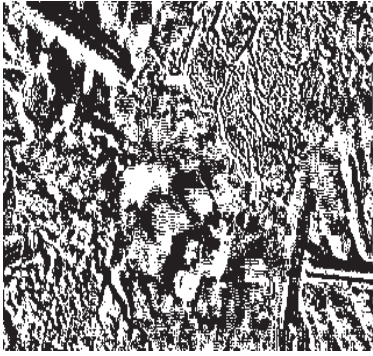


Fig 5.3(e)
Error image

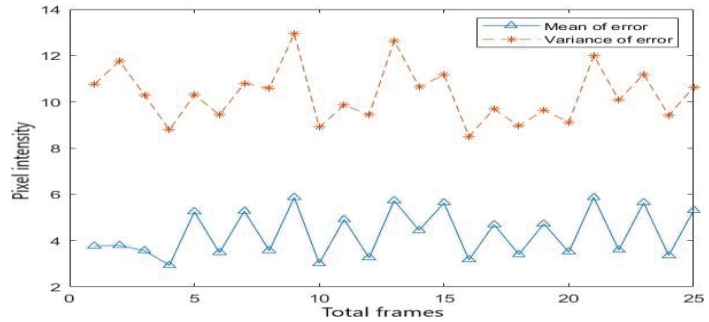


Fig 5.3(f)
Graphical representation of total frames and pixel intensity

Fig 5.3 video-1 with low pixel resolution

Here the error image is calculated for one frame by taking the difference between the input frame (greyscale frame) and output frame, as shown in Fig 5.3(e). So, by using this error image the mean error values and variance error values are calculated for a single frame of the input frame. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph, as shown in Fig 5.3(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity.

Now the pixel resolution of Fig 5.4 is increased to observe the effects that might occur in the video.

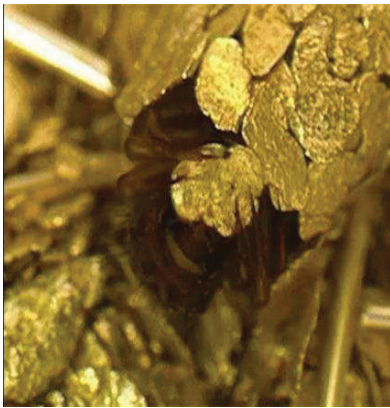


Fig 5.4(a)
Original image



Fig 5.4(b)
Greyscale frame



Fig 5.4 (c)
Encrypted frame

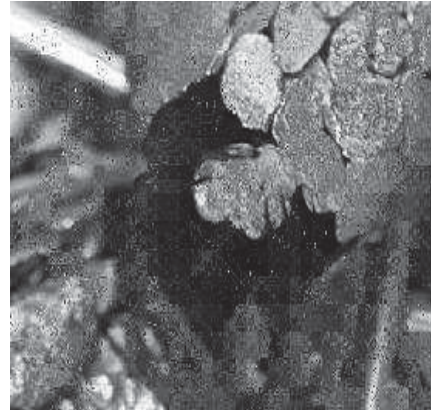


Fig 5.4 (d)
Output frame

The pixel resolution of the video used is 480×640 . From Fig 5.4 (a), the original frame which is taken from the video, the input frame(greyscale frame), encrypted frame, and output frame is shown in Fig 5.4 (b)(c)(d). The input frame and the output frame are used for the calculation of mean error and variance error values.

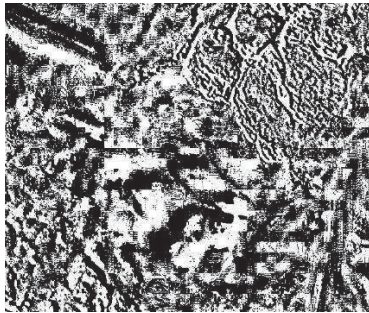


Fig 5.4(e)
Error image

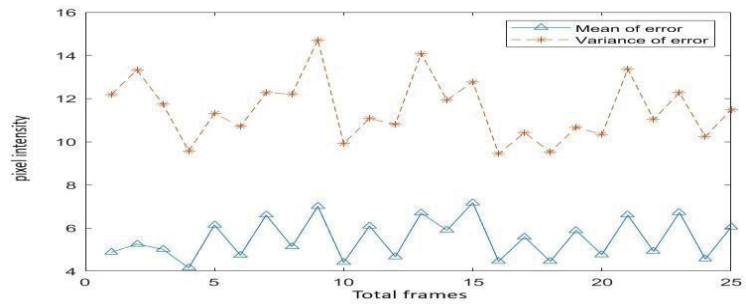


Fig 5.4(f)

Graphical representation of total frames and pixel intensity

Fig 5.4 video-1 with a high pixel resolution

Here the error image is calculated for one frame by taking the difference between the input frame (greyscale frame) and output frame, as shown in Fig 5.4(e). So, by using this error image the mean error values and variance error values are calculated for a single frame of the input frame. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph as shown in Fig 5.4(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity.

Here the comparison is done on all the three videos of different pixel resolutions, which are presented above. As compared to the error images of all the pixel resolutions, we can notice some changes with respect to the graph, output frame, and encrypted frame. The H.264 codec has some of the pixels mismatched at the output. So that is the reason the noise can appear a little bit in the video than original resolution. As the resolution increases, the higher the noise in the output frame. This can be observed from the results. The graph of mean error values and

variance error values has lots of degradation when compared to the original frame. And in the obscuring method, there is a bit of degraded output image when compared to the input frame. These are the main reasons to get issues in the output when both the codes are combined. And the lowest pixel resolution has higher security when all the encoded frames are compared to each other in all pixel resolutions. So, the scaling influences very little on the method used. In the lower and original pixel resolutions, the encrypted frame is in such a way that no one can even guess the information in it. As H.264 video codec doesn't provide greater compression to the higher resolution, there are lots of issues in the encrypted image that can reveal a bit of information. So when the scaling factor is compared between the pixel resolution, the graph and the error image observed show a very less difference between the mean error values to the variance error values in the graph. This perfectly suits the original pixel resolutions when compared with higher pixel resolutions. In higher pixel resolutions, the issues are more, and the error image has more differences. Finally, the scaling factor doesn't have an effect on lower and original pixel resolutions but not on higher resolutions.

5.3 Texture factor

Texture factor refers to different types of shapes of information in the video. The change in the information/shape and the movement can affect the feasibility or the quality of the method used. Here three different textured videos from Fig 4.19, 4.20, 4.21 are chosen to generate the results. These three videos have different types of information and movement to check whether any type of information or the movement had any effect on the method used. Here every video which is processed is shown and the statistical calculation like mean error values and variance error values are taken to see the difference between the input frame(greyscale frame) and output frame of all the frames and are plotted are shown in the graphical representation. So the process starts to have the video and is divided into frames. These are stored and for every frame, the input frame and output frame, the mean error values, and variance error values are calculated.

From Fig 5.5, video-2 is darker in color and the information is high. The movement is also high. As in Fig 5.5(a), the original frame can be seen with darker and lots of information. This video is divided, and the 1st frame is used to show the results. Fig 5.5(b) is the input frame (greyscale image of the first frame of the video). Then the encrypted frame from Fig 5.5(c) is used to send for transmission. The output is the frame after processing the method shown in Fig 5.5(d).



Fig 5.5(a)
Original frame



Fig 5.5(b)
Greyscale frame



Fig 5.5(c)
Encrypted frame



Fig5.5(d)
Output frame

From the 1st frame, these are the outputs generated. By using the input frame and the output frame, the difference can be calculated.



Fig 5.5(e)
Error image

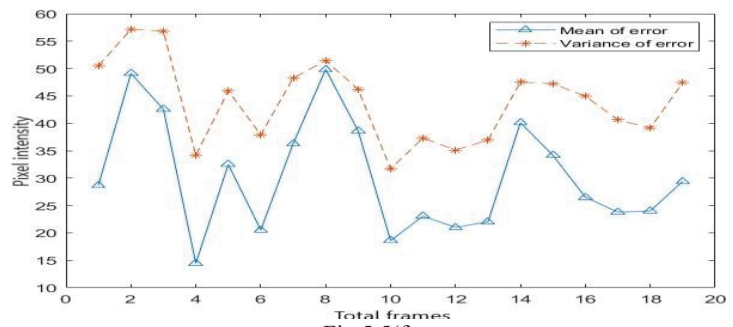


Fig 5.5(f)
Graphical representation of total frames and pixel intensity

Fig 5.5 video-2 with original pixel resolution

Here the error image is calculated for one frame by taking the difference between the input frame (greyscale frame) and output frame, as shown in Fig 5.5(e). So, by using this error image, the mean error values and variance error values are calculated for a single frame of the input frame. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph as shown in Fig 5.5(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity.

As of now, we are selecting another textured video that is video-3. The video-3 is a lighter color and taken as the medium information as it has only two objects on both ends that have the movement, as shown in Fig 5.6(a). Here this video is processed in the method and the outputs are shown in Fig 5.6(d).



Fig 5.6(a)
Original frame



Fig 5.6(b)
Greyscale frame

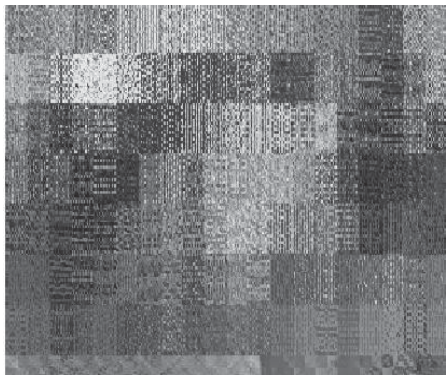


Fig 5.6(c)
Encrypted frame



Fig 5.6(d)
Output frame

From Fig 5.6(a) the original frame of the video-3 is shown. Then the frame is changed to greyscale and taken as the input frame of the 1st frame of the video, as shown in Fig 5.6(b). This continues with the encryption frame that is sent for transmission from Fig 5.6(c). From Fig 5.6(d), the output frame is the processed frame of the 1st frame of the video. From these outputs, the input frame and the output frame are used to calculate the mean error and variance error values of the frames.

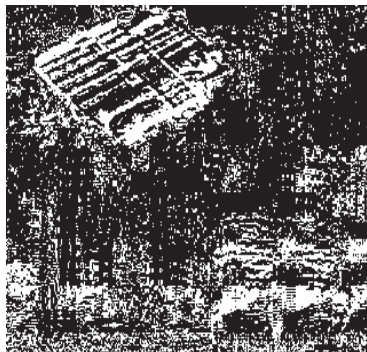


Fig 5.6(e)
Error image

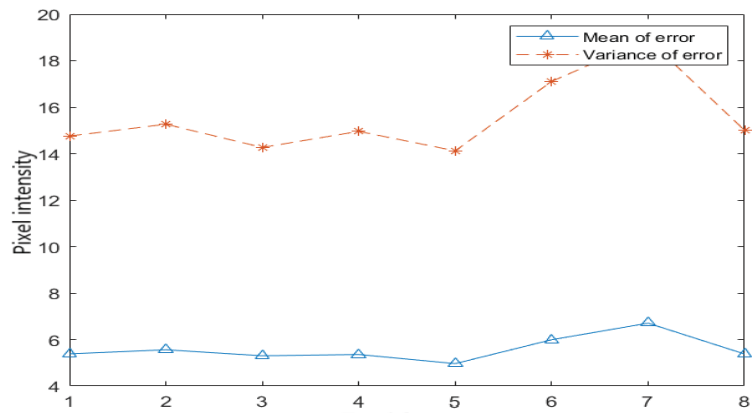


Fig 5.6(f)

Graphical representation of total frames and pixel intensity

Fig 5.6 video-3 with original pixel resolution

Here the error image is calculated for one frame by taking the difference between the input frame (greyscale frame) and output frame as shown in Fig 5.6(e). So, by using this error image the mean error values and variance error values are calculated for a frame of the input frames. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph as shown in Fig 5.6(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity. Finally, we are taking video-4 for the execution. The video-4 is a plain dark colored video. The plain textured video is chosen to the effect of the method in this textured.

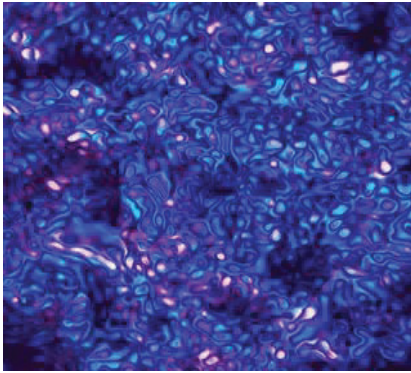


Fig 5.7(a)
Original frame

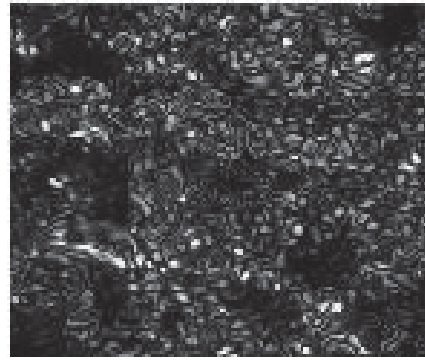


Fig 5.7(b)
Greyscale frame

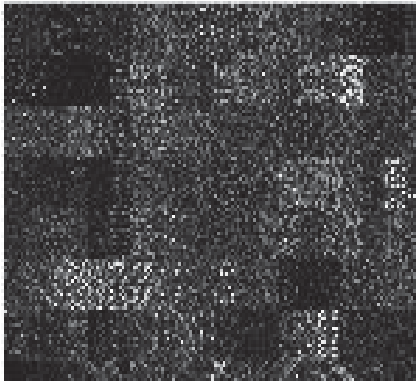


Fig 5.7 (c)
Encrypted frame

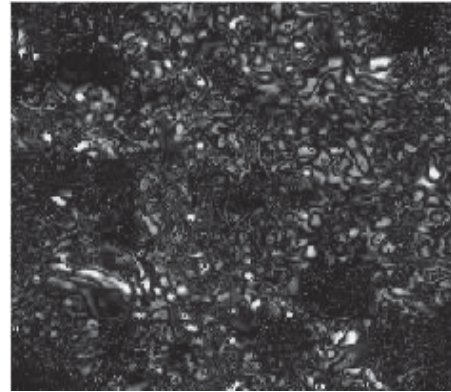


Fig 5.7 (d)
Output frame

Fig 5.7(a) shows the original frame that consists of a plain texture. Fig 5.7(b) shows the input frame of the 1st frame of the video. So, then the encrypted frame is shown in fig 5.7(c). And the output frame is the final output from Fig 5.7(d). By using the input frame and the output frame the mean error values and the variance error values are calculated and plotted.

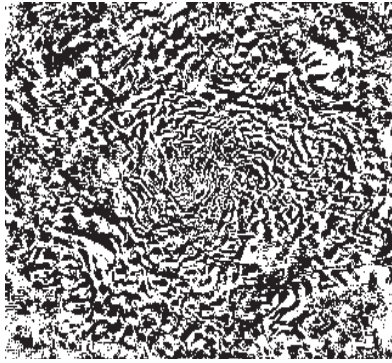


Fig 5.7(c)
Error image

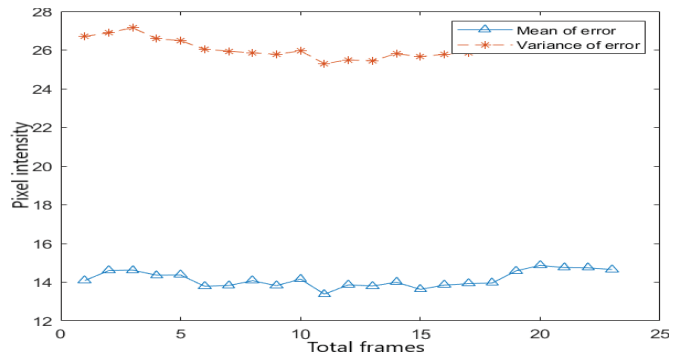


Fig 5.7(f)
Graphical representation of total frames and pixel intensity

Fig 5.7 video-4 with original pixel resolution

Here the error image is calculated for one frame by taking the difference between the input frame (greyscale frame) and output frame as shown in Fig 5.7(e). So, by using this error image the mean error values and variance error values are calculated for a frame of the input frames. Then these are calculated for all the frames in the video. Finally, the calculated values are plotted in a graph as shown in Fig 5.7(f). The pixel intensity is measured for the mean error values and variance error values of all the frames and is plotted. So, the mean error values of frame pixel intensity are compared with the variance error values of frame pixel intensity.

Here the videos are selected as per the texture and the movement of the information on the video. So, they selected three different textured videos with one lighter colored and two darker colored videos. When it comes to texture, three different informative videos are taken, as seen from the original frames of all the outputs shown. When observed from the outputs of the videos and the error images of all the videos, two of the dark colored videos have very less amount of noise in the output. This shows that the texture factor has a good impact on the method used as the mean error values and variance error values of the outputs have very little difference, which shows a good result. But the lighter colored video has lots of noise that show that the frame is a bit disturbed by the lighter color and the information in it. The video with brighter color has better outputs as the noise can't be seen more in it. So, from this analysis, the texture factor has a little bit drawback with light colored videos, but with darker videos and lots of information has less effect and less drawback and are clearer when compared to the lighter colored. This shows that texture has an impact on the information in the video.

This chapter discusses the main goal of our thesis, i.e., to encode the video that is encrypted using the obscuring method. This chapter also emphasizes the result obtained from the experimentation. The solution chosen for solving our research problem, as mentioned in chapter 1, was to encode the video that is encrypted by obscuring method. The encoding is done by the streaming video codec, so if this works, then this method can be used in streaming. And here, this thesis proves that it achieves to encode the video that is encrypted.

This thesis of encoding follows a 2 step process, so in the first step, the obscuring method is used to encrypt the video so that the video looks like noise. And in the second, this video which looks exactly like a noise needs to be gone through the encoding process and the compression of it is done in the end and then this video is sent to the decryption side to decrypt. On the decoding side, after decryption the results are good, so the performance analysis is done on the method used. This analysis is done in a way by taking different video pixels and different textured videos. Then this chapter discusses the research questions.

RQ1: What is the feasibility of the system while using obscuring method?

Answer: The obscuring method is as feasible as it was thought to be. From the analysis, we found out that the lower and original pixel resolutions videos have a lot of positives. As we observe from the results of the scaling factor and texture factor, there are no issues with the output of the video and even the security is satisfactory. As it can be observed that noise is very low in the lower resolutions when compared to higher resolutions below Fig (6.1). These are arranged from low to high resolution to observe the noise. Not only is the noise factor is seen here, but also the security factor can be seen from the pixel resolution videos. And this can also be observed in Fig (6.2). These are arranged from low to high resolution to see the difference with the higher resolutions. Here the security in higher resolutions is a bit revealing. But the security in the lower and original pixel resolution is very high. As nothing can be seen from the frame. Suppose something needs to be more securely delivered to anyone. Then the resolution doesn't have any use. Even the smaller resolution videos can be sent if security is essential, as even the low resolution videos can have the information that is in the higher resolution. So here, this shows that the method used is feasible in scaling factor. Just the concern is with the security in higher resolutions.

When coming to the texture factor, even the frame with more texture and more movement has good security as its information cannot be seen. And even the noise in the output frame of the darker textured video has less noise in the output frame. So even for the textured factor, the feasibility is surprisingly good. But the codec that is used in this thesis has a bit of an issue. While experimenting with the output of H.264 codec executing individually without any security, the output has some pixel mismatching. And in the security method without using H.264, the output image is a little bit more degraded than the input. So, in the output frames, there is a bit of noise that appears at the end of the decoding part. This method is feasible to achieve encoding the encrypted video. Also, at the same time, for security, the resolution is not so important. So, the lower and original resolution videos can be used as it is more secure when compared to higher resolution videos.

when coming to the feasibility, the main factor is to check the performance of the system. The performance depends on the memory, time of encoding, time of encrypting, quality of the video. Firstly, the memory is not an issue, and it is really feasible that encoding the encrypted video doesn't occupy much space. When compared with the original video, the encoded video has occupied a bit more space that it doesn't affect the performance of the system in any of the ways. The performance of time of encoding is a bit degraded. The reason behind this degradation is that it should encode the encrypted frame, and this encrypted frame is just a noise. So, the prediction takes time to predict between two noise frames and this degrades the system a bit. While moving to the performance of time of encrypting, it is better when compared to the time of encoding. It doesn't even take a minute to completely encrypt the frame into noise. So, the performance of this is not even a problem to the system. The quality of the video can be measured by checking the noise and comparing the input and output frames. So, when coming to the noise, in the output frames of every pixel only the highest pixel has some noise issues. So, this is not much issue to the system that it effects the performance. As this performance is checking for the system to work properly with the video and its properties. Then coming to the comparison of the input and output frame, there is not much difference in both the scaling and texture factors of the frames. And, there is no loss in the output frame as it is same as the input when observed in the results that are in the fifth chapter. So, the performance of the system is good, and it doesn't even arise any problem with the video or the encoding or encrypting process this thesis follows.

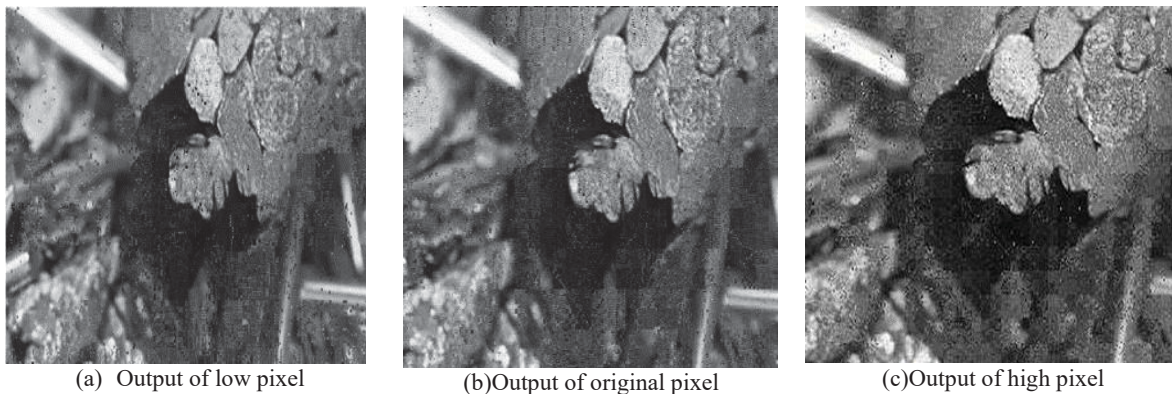


Fig 6.1: Noise comparison of different pixel resolutions

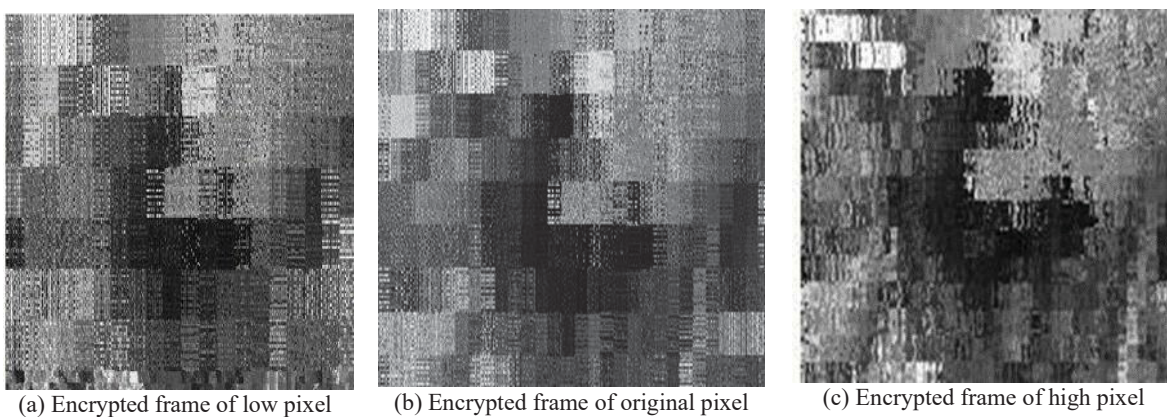


Fig 6.2: Encrypted frames comparison of different pixel resolutions

RQ2: How stable is the system having the obscure video into the encoded format?

Answer: In this, the obscuring method is used in the video to encrypt it and then encode and compress it with the H.264 codec. Here when the system stability is checked, it's excellent. The stability check is done based on the security of the video and the time taken by the video to encrypt it. From the generated results, we can see that the security of the videos is very high as the information in the frame is not visible to even guess what it has. Only in the higher pixel resolution, the security is a bit problem some of the information is revealing. And even with the textured factor, the security is high as we have used the original pixel resolution videos. As the error image is the difference between the input and output frame. There is not much difference in all the error images. Here the error and the graph have a high peak with the higher pixel resolution videos. Here the security of the video balances the stability of the system. When coming to time consumption is a crucial aspect of system stability. It takes very little time to obscure the video as it obscures the block, which has very little information. Even the texture factor doesn't affect the delivery with either the darker or lighter colored videos. So hereby the discussion, the stability of the system is satisfactory while encoding the encrypted video.

So, the feasibility of the system is high, the computational speed increases, the quality of the frame in the output is good. And the stability of the system remains satisfactory when it comes to security or delivery. This discussion shows what precisely the thesis is expected to do and the expectation meets the requirements.

Conclusion and Future work

Conclusion

Most people are now streaming online. When it comes to streaming, video streaming outperforms other things. One of the concerns that come with streaming is security. Security is the most needed act to secure the content, not to being hacked or copied. So to secure video streaming, there are security methods that use key, protocol, or some other secure web hosting sites. But the idea of this thesis is to encrypt the video without any aid and that even the video is hacked, it cannot be understood. So for that, an obscuring method to encrypt the video is used. This method just encrypts the video with the information in it and makes it look like noise. And then, this encrypted video is encoded and compressed with the help of the streaming video codec. Here H.264 codec is used for this thesis as it is widely used and is compatible with almost all devices. So this thesis firstly is to check whether it is possible to interrupt the codec and do the obscuring method, and then the rest of the encoding and compression is done with the video codec. And is it possible to encode the encrypted video is checked? So, the results show that it is possible to encode the encrypted video. Hereafter the thesis focuses on the system performance while using this obscuring method. This is done by the performance analysis with the scaling and texture factor from the results and analysis. These factors are checked by having different type of videos and different pixel videos and checking it accordingly. And even the mean error values and variance error values are taken for every frame of the video so as to observe the difference between the input and output frames of the video. These values are plotted with the pixel intensity to check the difference. So from the discussion part, it shows that the feasibility of the system is good as it provides good security to the videos and the noise is less in the output. When coming to stability, the delivery of the video is good, and it is checked by the compilation time of the video to encrypt. This aspect is also discussed in the earlier chapter that it consumes very little time when obscuring method is used. So from the results and discussion, this concluded that the thesis successfully tried to get the required results. And it shows that the obscuring method can be used to encrypt the video, and it shows that the video is secured by using this method. And it also concludes that the encrypted video can be encoded by using a video codec. From the performance analysis, it concludes that lower and original pixel resolution videos are with very little difference from the input to output and the security is more. Even the darker colored and with more motion videos are good at securing the information. So, this method is perfect for use in streaming.

Future work

When discussing the further implementation to the thesis, there is a lot of scope. In this thesis we explored that is it possible to secure the video and to use this in streaming. we also show the possibilities and the drawbacks of the thesis. For the further process, the thesis can be extended by using the protocol in the streaming and the video which is encrypted can be streamed by using the streaming protocol. So, by doing this task, the streaming is done securely without any key or any external device of the encoded frame. The output of the frame has a little bit of noise and some degradation. So, from this the image quality can be improved. Here we can also do the quality of experience regarding the streaming and we have a scope to do the security measures to check the feasibility.

References

- 1) Streaming introduction available at <https://www.theverge.com/2020/3/27/21195358/streaming-netflix-disney-hbo-now-youtube-twitch-amazon-prime-video-coronavirus-broadband-network>
- 2) Streaming introduction available at <https://www.forbes.com/sites/markbeech/2020/03/25/covid-19-pushes-up-internet-use-70-streaming-more-than-12-first-figures-reveal/?sh=26f2b9df3104>
- 3) Streaming importance available at <https://www.weforum.org/agenda/2021/03/streaming-service-subscriptions-lockdown-demand-netflix-amazon-prime-spotify-disney-plus-apple-music-movie-tv>
- 4) Video streaming stats and facts available at <https://www.comparitech.com/tv-streaming/streaming-statistics/>
- 5) Holankar, D., & Stamp, M. (2004, January). Secure streaming media and digital rights management. In *Proceedings of the 2004 Hawaii International Conference on Computer Science* (pp. 85-96).
- 6) Giradkar, Shrutika S., and Antara Bhattacharya. "Securing compressed video streams using RC4 encryption scheme." *2015 Global Conference on Communication Technologies (GCCT)*. IEEE, 2015.
- 7) Live streaming available at <https://www.wowza.com/blog/complete-guide-to-live-streaming>
- 8) Sullivan, Gary J., Pankaj N. Topiwala, and Ajay Luthra. "The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions." *Applications of Digital Image Processing XXVII*. Vol. 5558. International Society for Optics and Photonics, 2004.
- 9) Asiya, Asif Ali Laghari Hui He, and Khan Sajida Karim. "Impact of Video File Format on Quality of Experience (QoE) of Multimedia Content." *development* 10: 11.
- 10) Iyyanar, P., M. Chitra, and P. Sabarinath. "Effective and Secure Scheme for Video Streaming Using SRTP." *International Journal of Machine Learning and Computing* 2.6 (2012): 855.
- 11) Ouchi, Samir, et al. "Security Estimation in Streaming Protocols." *2011 International Conference on Innovations in Information Technology*. IEEE, 2011.
- 12) Wang, Chia-Hui, and Jan-Ming Ho. *Application-layer security control for real-time video streaming*. Technical report, Institute of Information Science, Academia Sinica, Taiwan, 2005.
- 13) S. C. Iyer, R. R. Sedamkar and S. Gupta, "A novel idea of video encryption using hybrid cryptographic techniques," 2016 International Conference on Inventive Computation Technologies (ICICT), 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7830094.
- 14) Lin, Eugene T., et al. "An overview of security issues in streaming video." *Proceedings International Conference on Information Technology: Coding and Computing*. IEEE, 2001.
- 15) Rajamanickam, Vani, and Sangeetha Marikkannan. "Performance evaluation of motion estimation in H. 264/AVC encoder." *Proceedings of the International Conference on Advances in Computing, Communications, and Informatics*. 2012.
- 16) Deshpande, Renuka G., and Lata L. Ragha. "Performance analysis of various video compression standards." *2016 International Conference on Global Trends in Signal*

- Processing, Information Computing and Communication (ICGTSPICC)*. IEEE, 2016.
- 17) Ponlatha, S., and R. S. Sabeenian. "Comparison of video compression standards." *International Journal of Computer and Electrical Engineering* 5.6 (2013): 549- 554.
 - 18) Sullivan, Gary J., and Thomas Wiegand. "Video compression-from concepts to the H. 264/AVC standard." *Proceedings of the IEEE* 93.1 (2005): 18-31.
 - 19) Li, Yan, and Main Cai. "H. 264-Based multiple security levels net video encryption scheme." *2009 International Conference on Electronic Computer Technology*. IEEE, 2009.
 - 20) Wiegand, Thomas, et al. "Overview of the H. 264/AVC video coding standard." *IEEE Transactions on circuits and systems for video technology* 13.7 (2003): 560-576.
 - 21) De Queiroz, R. L., Ortis, R. S., Zaghetto, A., & Fonseca, T. A. (2006, September). Fringe benefits of the H. 264/AVC. In *2006 International Telecommunications Symposium* (pp. 166-170). IEEE.
 - 22) Po, L. M., & Ma, W. C. (1996). A novel four-step search algorithm for fast block motion estimation. *IEEE transactions on circuits and systems for video technology*, 6(3), 313-317.
 - 23) File containers available at <https://dev.to/hamitdemir/what-is-a-video-container-format-41e8>
 - 24) Video codec history available at <https://medium.com/@JackPu/how-js-get-video-codec-548a33cf7454>
 - 25) Streaming process available at https://miro.medium.com/max/1050/1*Qb9cz8TRaUf7EhLgzGtNDw.png
 - 26) Fanny Lalonde Levesque, Jude Nsiempba, José M. Fernandez, Sonia Chiasson, and Anil Somayaji. 2013. A clinical study of risk factors related to malware infections. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13)*. Association for Computing Machinery, New York, NY, USA, 97–108. DOI:<https://doi-org.miman.bib.bth.se/10.1145/2508859.2516747>
 - 27) E. T. Lin, G. W. Cook, P. Salama and E. J. Delp, "An overview of security issues in streaming video," *Proceedings International Conference on Information Technology: Coding and Computing*, 2001, pp. 345-348, doi: 10.1109/ITCC.2001.918819.
 - 28) Hartsell, T. & Yuen, S.C.Y. (2006). Video Streaming in Online Learning. *AACE Review* (formerly *AACE Journal*), 14(1), 31-43. Chesapeake, VA: Association for the Advancement of Computing in Education (AACE). Retrieved January 21, 2022 from <https://www.learntechlib.org/primary/p/6152/>.
 - 29) Apostolopoulos, J. G., Tan, W. T., & Wee, S. J. (2002). Video streaming: Concepts, algorithms, and systems. HP Laboratories, report HPL-2002-260.
 - 30) Golston, J. (2004). Comparing media codecs for video content. In *Embedded Systems Conference*, San Francisco.
 - 31) Amritpal Kaur and Sarbjeet Singh. 2021. A Survey of Streaming Protocols for Video Transmission. In *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence (DSMLAI '21)*. Association for Computing Machinery, New York, NY, USA, 186–191. DOI:<https://doi-org.miman.bib.bth.se/10.1145/3484824.348489>
 - 32) Gloe, T., Fischer, A., & Kirchner, M. (2014). Forensic analysis of video file formats. *Digital Investigation*, 11, S68-S76.
 - 33) MWV available at <https://bytescout.com/blog/2018/02/windows-media-video-format.html>
 - 34) Explanation of frames and blocks available at <https://programmersought.com/article/41525042885/>
 - 35) Ahirwal, B., Khadtare, M., & Mehta, R. (2007, November). FPGA based system for color

- space transformation RGB to YIQ and YCbCr. In *2007 International Conference on Intelligent and Advanced Systems* (pp. 1345-1349). IEEE.
- 36) FBMC and VBMC available at https://www.researchgate.net/figure/The-choosing-of-the-block-size-the-fixed-size-block-matching-FSBM-and-the-variable_fig2_261241025
 - 37) Uppala, Pavan Kumar, and Bharath Sangasani. "Implementation of Recursive method in Image Steganography." (2015).
 - 38) Reference frame and motion compensation frame available at <https://answers.opencv.org/question/81353/motion-estimation-between-2-frames/>
 - 39) Block size in H.264/AVC available at https://www.researchgate.net/figure/The-various-block-sizes-in-H264-AVC_fig1_220845531
 - 40) Intra frame prediction available at <https://www.vcodex.com/h264avc-intra-precision/>
 - 41) Image compression model available at https://www.researchgate.net/figure/General-Compression-Model_fig1_285110254
 - 42) Venčkauskas, Algimantas & Morkevicius, Nerijus & Bagdonas, Kazimieras & Damasevicius, Robertas & Maskeliunas, Rytis. (2018). A Lightweight Protocol for Secure Video Streaming. *Sensors*. 18. 10.3390/s18051554.
 - 43) P. J. Criscuolo, "Distributed Denial of Service Tribe Flood Network 2000 and Stacheldraht CIAC-2319 Department of Energy Computer Incident Advisory Capability (CIAC) UCRL-ID-136939 Rev. 1.", Lawrence Livermore National Laboratory, February 14, 2000.
 - 44) Copyright infringement available at <https://www.investopedia.com/terms/c/copyright-infringement.asp>
 - 45) R. S. Meurer, T. R. Mück and A. A. Fröhlich, "An Implementation of the AES Cipher Using HLS," *2013 III Brazilian Symposium on Computing Systems Engineering*, 2013, pp. 113-118, doi: 10.1109/SBESC.2013.36.
 - 46) DRM available at [https://www.widen.com/blog/digital-rights-management#:~:text=Digital%20rights%20management%20\(DRM\)%20is,prevent%20unauthorized%20modification%20or%20distribution.](https://www.widen.com/blog/digital-rights-management#:~:text=Digital%20rights%20management%20(DRM)%20is,prevent%20unauthorized%20modification%20or%20distribution.)
 - 47) Wee, Susie J., and John G. Apostolopoulos. "Secure scalable video streaming for wireless networks." *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*. Vol. 4. IEEE; 1999, 2001.
 - 48) Y. Chen et al., "An Overview of Core Coding Tools in the AV1 Video Codec," *2018 Picture Coding Symposium (PCS)*, 2018, pp. 41-45, doi: 10.1109/PCS.2018.8456249.
 - 49) D. Mukherjee *et al.*, "The latest open-source video codec VP9 - An overview and preliminary results," *2013 Picture Coding Symposium (PCS)*, 2013, pp. 390-393, doi: 10.1109/PCS.2013.6737765.
 - 50) S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," *Proceedings IEEE International Conference on Communications ICC '95*, 1995, pp. 1778-1783 vol.3, doi: 10.1109/ICC.1995.524505.
 - 51) G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.
 - 52) Frame division block available at https://www.researchgate.net/figure/Divide-the-reference-image-into-blocks-and-predict-the-corresponding-block-in-the-sensing_fig3_324811999
 - 53) Booth, W., Noras, J. M., & Xu, D. (1998, January). A novel fast three-step search algorithm for block-matching motion estimation. In *Asian Conference on Computer Vision* (pp. 623-

- 630). Springer, Berlin, Heidelberg.
- 54) Liu, K., Qiu, Q., & Zhang, Z. (2011, July). A novel fast motion estimation algorithm based on block-matching. In *Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference* (Vol. 2, pp. 1402-1405). IEEE.
- 55) Po, L. M., & Ma, W. C. (1996). A novel four-step search algorithm for fast block motion estimation. *IEEE transactions on circuits and systems for video technology*, 6(3), 313-317.