An MPEG-4 based Selective Encryption Algorithm for Secure Real-time Video Streaming

Jainish Shah
Alpha Circuit LLC
730 N Oaklawn Ave,
Elmhurst, Illinois, USA

Nilanjan Sen
Western Illinois University
1 University Circle
Macomb, Illinois, USA

Binto George
Western Illinois University
1 University Circle
Macomb, Illinois, USA

Antonio Cardenas-Haro
Western Illinois University
1 University Circle
Macomb, Illinois, USA

ABSTRACT

Real-time video encryption is a critical need in today's digital era. With the proliferation of online activities like streaming, virtual meetings, and transmission of sensitive data, the demand for secure data transmission is at an all-time high. In this paper, a new algorithm is proposed for encrypting MPEG-4 video frames. This algorithm involves encrypting the DCT values of a macroblock XOR-ed with the encryption key to significantly enhance the encryption speed and robust security while reducing the computational cost.

Keywords

Selective encryption, Video encryption, Real-time video, Real-time streaming

1. INTRODUCTION

The importance of video live-streaming for official and entertainment purposes is undebatable. The usage of video live-streaming has increased profusely during and after the COVID-19 pandemic. Video conferencing services like Google Meet and Zoom have become increasingly prevalent worldwide. This is because these are the effective ways businesses, government organizations, and individuals can follow to conduct meetings and other activities remotely. Therefore, it is crucial to evaluate cybersecurity-related issues with these systems. Since cybercriminals infiltrated several organizations' computer systems in the past, the situation has significant financial ramifications for digital media security.

The original data that needs to be communicated or saved is referred to as plaintext, and it can be read and understood by humans. The plaintext must be encrypted (a.k.a. ciphertext) to make it unreadable to unauthorized people. The authorized recipient of the message decrypts the ciphertext to convert it to plaintext. A cryptosystem refers to a system or item that offers encryption and decryption [2]. An encryption algorithm's security level is measured by its key space size [14]. Popular encryption schemes, such as Data Encryption Standard (DES), Rivets-Shamir-Adelman (RSA) algorithm, and Advanced Encryption Standard (AES) are not effective for encrypting videos because they are typically slow and demand a lot of computing power [3].

This paper presents a comprehensive study of the selective encryption method for real-time video encryption and proposes a selective encryption algorithm for MPEG-4 video frames. Selective encryption is a technique that encrypts only the most sensitive parts of the video. The study is meticulously done to evaluate the performance of this method in terms of security, video quality, and transmission delay. The challenges and trade-offs of selective encryption in real-time video are thoroughly examined. While numerous schemes, including full encryption, permutation-based encryption, perceptual encryption, and hybrid encryption, have been proposed for video encryption, selective encryption schemes could be highly effective and deserve more attention.

The rest of the paper is organized as follows. In Section 2, the basic concepts of video encryption are discussed briefly. Section 3 provides a discussion of the related works. Section 4 and Section 5 discuss the motivation and the research methodology respectively. The proposed algorithm is discussed in section 6. Results and analysis are discussed in Section 7. Lastly, Section 8 contains the conclusion along with the future work.

2. BACKGROUND INFORMATION

This section emphasizes background information on video encryption. The confidentiality and integrity of video data must be guaranteed, given the rapid evolution of digital technologies. Using cryptographic methods, video encryption algorithms protect the content of video files. For online conferencing, healthcare, and online education, real-time video encryption techniques are especially crucial since the data must be processed and transferred in real-time. While minimizing latency and preserving good video quality, these algorithms must be able to encrypt and decode video data instantly. Video encryption algorithms can be implemented using various cryptographic methods, including symmetric key encryption, public key encryption, and hash functions.

There are two levels of protection for digital images: low-level and high-level security encryptions. In low-level security encryption, the visual quality of the encrypted image is lower than that of the original, but the viewers can still see and interpret the image's content. The image seems random noise in the high-level security situation, and the content gets entirely scrambled. In this instance, viewers cannot in any way watch the video. Selective encryption ensures confidentiality without encrypting every bit of a digital image. So, for better performance, it's important to encrypt only a tiny portion of the bitstream [5].

High-definition video, interactive multimedia, and digital rights management employ the most modern standard, which is MPEG-4. It uses a more sophisticated compression technique that can outperform MPEG-2 in terms of compression ratio. A frame inside a video file is divided into three frame types: I-frames, P-frames, and B-frames. I-frames (Intra-frames) are autonomous frames that can be decoded without using other frames for reference and hold the entire image. P-frames are frames projected based on earlier I or P frames. Bidirectional frames, or B-frames, may be anticipated from the current and the next I or P frames. According to the MPEG-1 video coding scheme, a video comprises a series of images known as Group of Pictures (GOPs). Each GOP includes I, P, and B frames [4].

Some of the video encryption algorithms are discussed below:

2.1 Selective encryption-based video encryption algorithm

This methodology encrypts only the crucial sections of the video instead of the entire video. It has been demonstrated to successfully safeguard sensitive information while preserving video quality and transmission speed.

2.2 Public key-based video encryption algorithm

This methodology incorporates public key encryption techniques, such as RSA and Elliptic Curve Cryptography (ECC), to safeguard video data. It has been demonstrated to offer a strong degree of security and to withstand attacks like brute force and differential attacks. One of the main disadvantages is that public key encryption algorithms, such as RSA and Elliptic Curve Cryptography (ECC), are computationally intensive, making them less suitable for real-time video encryption.

2.3 Hybrid encryption-based video encryption algorithm

This methodology employs a mixture of encryption techniques, including chaotic maps and block ciphers, to secure video data. Multiple encryption methods have been demonstrated to offer a superior level of protection compared to relying on a solitary encryption technique.

2.4 Stream cipher-based video encryption algorithm

This method encrypts video data using stream ciphers like RC4 and Salsa20. It has been demonstrated to be very secure and efficient in calculation time. Stream ciphers are sensitive to errors in transmitting encrypted video data, which can lead to errors in the decrypted video. Furthermore, Stream cipher-based video encryption may not be reliable over time because of its sensitivity to the original parameters and conditions.

2.5 Block cipher-based video encryption algorithm

This method encrypts video data using block ciphers like AES and DES. It has been proven efficient at preventing illegal access to video data. Despite being widely utilized, it has several restrictions. Block ciphers' susceptibility to assaults such as known plaintext attacks, ciphertext-only attacks, and differential attacks is one of their fundamental drawbacks. Block ciphers can also lose security if hackers develop new ways to crack them.

2.6 Chaos-based video encryption algorithm

In this method, video data is encrypted using chaotic maps. It has been demonstrated to offer a high level of security and to be impervious to assaults like differential and brute-force attacks. The significant drawback of chaos-based video encryption is its computational complexity, which makes real-time video encryption less feasible. Second, the chaotic systems employed in video encryption are predictable, which makes them less secure because they can be foreseen over time. The size of the key necessary for chaos-based encryption can also increase the complexity and overhead of the encryption process.

3. RELATED WORK

Video data security is becoming increasingly important with the widespread use of the H.264 standard. An analysis of existing literature reveals that 17% of authors employed naive or full encryption, 9% utilized permutation-based encryption, and 51% implemented selective encryption, making it the most popular approach. Furthermore, 17% of authors explored perceptual encryption, while only 6% adopted hybrid encryption, indicating a clear preference for selective methods in video encryption research [14].

A paper from Yajun Wang et al. [20] presented a new selective encryption scheme for H.264-based video data security. Their scheme combined AES OFB mode with a sign encryption algorithm to encrypt DCs and parts of ACs. They claimed that it was secure, had low complexity, and also supported error-propagation prevention. Moreover, due to little effect on compression ratio, it was also suitable for secure mobile and wireless multimedia transmission.

Spanos and Maples [17] proposed a mechanism known as Aegis. It applied the video compression method to reduce the video image size, hence requiring less video data to be encrypted. They tested this mechanism using three types of video traffic to compare the delay performance and queue requirements.

There have been various attempts to safeguard video data, but these methods have limitations or cause significant delays. Liu et al. [12] proposed a security system for the MPEG video compression standard. Their approach involved DCEA (DC Coefficient Encryption Algorithm) and "Event Shuffle." The first method encrypted a group of DC coefficient codewords using data permutation to scatter the ciphertexts of additional codes in it. These additional codes were encrypted beforehand using the block cipher. On the other hand, the event shuffling encrypted only the DC coefficients. This method shuffled the AC events that were generated after DCT transformation and Quantization. They found that their methods did not increase the bit overhead of the MPEG bit stream and had low processing overhead for the MPEG codec, as shown by their experimental results.

Thomas et al. [18] proposed an H.264 selective encryption algorithm that encrypted transform coefficients' sign bits and motion vectors. They claimed that this encryption scheme was better than I-frame encryption and that the algorithm complexity was lesser too. They also stated that their algorithm paved the path for transcoding independent of decryption terms.

A survey paper from Shah and Saxena [9] contained numerous video encryption algorithms based on DES/ IDEA that were available for secure MPEG video encryption. They classified the video encryption algorithms into standard cryptography algorithms, selective algorithms, permutation algorithms, and perceptual algorithms. They evaluated the algorithms' performance based on six parameters, viz. Visual Degradation, Encryption Ratio, Speed, Compression Friendliness, Format Compliance, and Cryptographic Security. They concluded that each type of algorithms had some pros and cons. So, an algorithm should be chosen depending on the applications' requirements.

Goel and Chaudhari [6] also worked on selective encryption. They proposed a median-based technique for selective image encryption, which aimed to reduce the heavy resource requirements on hardware platforms due to the large size of image data. The technique divided the image into blocks and determined the pixels to encrypt based on the median value of each block. This approach significantly reduced the amount of encrypted data and the encryption time. It kept track of encrypted and unencrypted pixels using a mask. Additionally, it was implemented by FPGA, leading to little overhead on area requirements.

Hofbauer et al. [8] used a hierarchical codec to evaluate a selective encryption approach for MPEG videos. The proposed approach encrypted nearly 0.3% for 125 P- or B-frames in a GOP to achieve severe distortion in the video stream that is hard to watch. By selectively encrypting certain portions of the video, their approach could provide a trade-off between security and compression efficiency while maintaining scalability.

Another selective encryption algorithm was proposed by Li et al. [11]. This algorithm randomly selected data using several pseudo-random sequences generated using RC4 with separate keys. So, no key information was required here for encryption. According to their claim, the computational cost of this scheme was less than 7 percent compared to the naive algorithm while keeping the video size compliant with the video codec and format.

Wang and Wang [19] proposed an encryption scheme that encrypts data based on different inter- and intra-prediction modes for the H.264 standard. This algorithm ensured format compliance, had a minimal impact on compression, and provided flexible security levels for different applications.

Apart from these, there were other selective encryption algorithms proposed by Sbiaa et al. [16] and Fei et al. [15]. They proposed encryption algorithms for the H.264 Advanced Video Coding (H.264/AVC) standard. Krikor et al. [10] proposed a selective image encryption algorithm using higher frequencies of DCT coefficients and stream cipher. Malladar and Kunte [13] proposed another selective video encryption based on the entropy measure of the blocks.

4. MOTIVATION

Studying and examining real-time video encryption security exploits is motivated by protecting video data. The real-time video encryption schemes must ensure data confidentiality from when it is transmitted from the source until the authorized parties receive it. This is crucial to preserving the safety of video data in fields such as surveillance, health care, and entertainment. Also, Cyberattacks and data breaches have increased due to the expansion of the internet and the growing usage of technology. These dangers have brought attention to the requirement for solid defenses against unauthorized access to video data. Furthermore, new approaches and techniques are continually being developed in the field of video encryption to increase the security of video data. The most challenging part of real-time video encryption is the massive volume of video data, especially in the era of Video on Demand (VoD). Research on real-time video encryption may seek to close the gap in the literature by thoroughly investigating a novel or under-researched encryption technique.

5. RESEARCH METHODOLOGY AND IMPLEMENTATION

Real-time video encryption is one of the areas where encryption is of utmost importance to maintain privacy and security. The increasing demand for secure communication has led to the development of various encryption techniques. This research proposes a real-time video encryption algorithm that utilizes an XOR operation on 11 bits of an unsigned integer and the sign bit (12 bits in total) of the first 6 DCT coefficients for 16×16 macroblocks, with the randomly generated 256-bit key using Mersenne Twister engine and key-dependent matrices. This algorithm aims to provide a secure and efficient encryption method that can be implemented in real-time video transmission systems.

The DCT is a widely used transformation technique in video compression, particularly for JPEG and MPEG. It effectively reduces spatial redundancy by converting spatial domain data into frequency domain data. This algorithm targets the most significant frequency components by changing the DCT components to conceal critical information of the video frames. Modifying these DCT values is advantageous because the coefficients represent the essential characteristics of the video content, such as luminance and chrominance. By encrypting these values, the algorithm disrupts the correlation between adjacent frames and hinders the ability of potential attackers to infer the original content through statistical analysis. This approach also leverages the inherent redundancy in video data. Since consecutive frames are often similar, selectively altering DCT coefficients can significantly enhance security while minimizing the computational load required for encryption.

5.1 Problem Statement

Video streaming has become essential to modern communication systems, including entertainment, education, and business. With the growing popularity of video streaming, the security of the transmitted video data has become a significant concern. Video encryption is a popular solution to ensure the confidentiality and integrity of video data during transmission. However, existing video encryption techniques have limitations regarding real-time video streaming. In real-time video streaming, the encryption and decryption process requires high processing power and low latency. As a result, it is crucial to have a practical solution and secure video encryption technique that can provide real-time video streaming with minimum processing overhead and latency. This work aims to propose and evaluate a selective video encryption technique that can ensure the confidentiality and integrity of real-time video streaming with minimum processing overhead and latency.

5.1.1 Configuration. All program implementation uses an HP laptop with 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz and 64-bit Windows 11 Home operating system with 16 GB RAM. Testing and programming specifically involved using Visual Studio 2022. LibAV, OpenCV, and OpenSSL open-source libraries are used in programming to achieve the output.

5.2 Video Processing

Video processing is a crucial component of a video encryption algorithm for which the LibAV and OpenCV are used. Video processing involves manipulating the video data to apply encryption algorithms to the video stream, ensuring the data is encrypted and decrypted.

5.2.1 LibAV. LibAV is an open-source library that provides tools for handling multimedia files. It can encode, transcode, and decode audio and video files, among other tasks. LibAV can help efficiently manage and process multimedia files and extract essential data from those files for analysis. It makes playing with the multimedia file's data accessible, manipulates and converts files to a different format, and extracts and analyzes the file's metadata for other purposes.

5.2.2 OpenCV. OpenCV (Open-source Computer Vision Library) provides comprehensive image and video analysis, processing, and manipulation tools. One of the main advantages of using OpenCV is its versatility and flexibility. It offers a range of pre-built functions and algorithms that can be easily customized and adapted to the research needs.

To further process the raw frame after decoding it, its values are converted to the matrix of float values to find the DCT values. The DCT values are passed to the encryption algorithm, where the bits of an integer are encrypted. The pass-by-reference functionality changes the original data with the newer values of the matrix.

6. PROPOSED ALGORITHM

This work is primarily centered on the H.264 format, a key component of Video on Demand (VoD). A practical solution is developed which is a hybrid encryption scheme that combines symmetric and asymmetric encryption techniques to protect H.264 encoded video data. This scheme, which uses a symmetric key generated using Data Encryption Standard, is directly applicable to real-world VoD systems, enhancing their security.

First, the frame is divided into multiple macroblocks of 16×16, which contain four blocks of 8×8 Y, one of 8×8 Cb, and one of 8×8 Cr. The frame format is in YUV (YCbCr) format, as shown in Figure 1. The image in Figure 1 has been taken from [4].

After making 8×8 blocks, each block is processed by three operations, which are Discrete Cosine Transformation (DCT), Quantization, and finally the Huffman Entropy Coding. DCT works on the lower spatial frequencies of the blocks. In the second phase of operations, many DCT coefficient becomes zero and the output is linearized as a zig-zag order. Finally, the Huffman Entropy Coding generates the compressed bitstream [4].

The proposed encryption algorithm uses an XOR operation on 11 bits and the sign bit of the first 6 DCT coefficients, with the bits of the generated key and key matrix. Here, the key matrix is generated based on the 256-bit keys for each channel. This adds more randomness and complexity. An attacker would have difficulty decrypting the data without the right keys and key matrices. The XOR is the fastest bitwise operation among the others and is helpful for bit manipulation. The algorithm encrypts MPEG-4 video slice by slice, as shown in Algorithm 1. XORing changes the bit of an integer and could give a different value than the original.

The DCT's output is the collection of 64 basis-signal amplitudes (also known as DCT coefficients). These Values play a crucial role in transforming amplitude of waveform to the coefficient values for the cosine function. The DCT coefficients are classified as ”DC coefficients” or ”AC coefficients.” These coefficients are part of the DCT values matrix. Manipulating these values can confuse the cosine function in regenerating the amplitude. The following 64 results are listed in a zig-zag fashion, starting with the DC coefficient and moving on to increasing frequency AC coefficients. The DC coefficient has a zero frequency in both dimensions, while AC coefficients are the remaining 63 with non-zero frequencies. In other words, most spatial frequencies have zero or near-zero amplitude and do not require encoding. The DC coefficient represents the average color that is present. The 63 AC coefficients represent color shifts inside the macroblock. Low-numbered coefficients indicate progressive color shift across the region or low-frequency color change. High-numbered coefficients indicate changes that occur frequently or when the color quickly shifts from one block's pixel to another. Lower-frequency AC coefficients are more critical than higher-frequency, and DC coefficients are more important than AC coefficients. The Inverse of DCT converts coefficients back to the spatial domain from frequency format. The classical 1-D DCT takes $O(N \log_2 N)$ time [7], where N is a block size of 8×8 = 64.

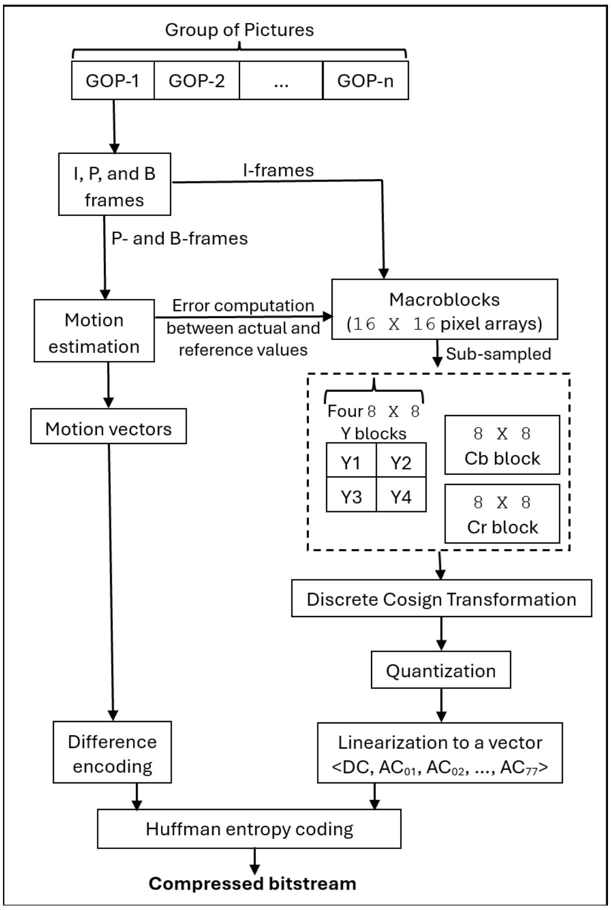

Fig. 1. MPEG video coding technique[4].

In order to illustrate the operations of the algorithm, The algorithm selects at most 1 DC and 5 AC coefficients from each macroblock in zig-zag pattern, as shown in Figure 2, of each channel using DCT. The Figure 2 has been taken from [1]. For all 11 bits of each coefficient and the sign bit (11 bits + 1 sign bit = 12 bits), XOR operation is performed by selecting random key index and key-matrix values. In an encryption/ decryption process, two different formulas are used to encrypt the bits and sign bits of coefficients to randomize the bit selection process out of the key and key matrix.

The first formula is used to encrypt each bit of the coefficients. It involves selecting 256-bit key and 256-byte key matrix values using the current column, row, and bit indices with specific offsets. These values are shifted to the right by 7 bits, then combined using the XOR operation. The final step involves bitwise ANDing the result

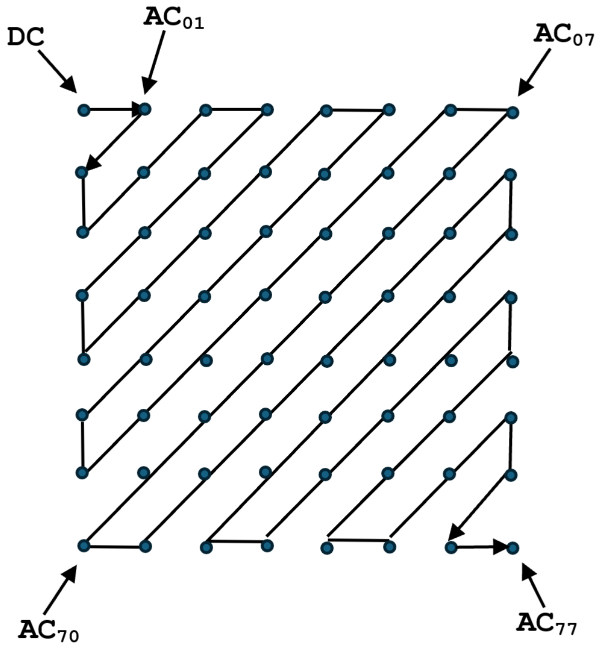

Fig. 2. The zig-zag sequence of the DCT coefficients order [1].

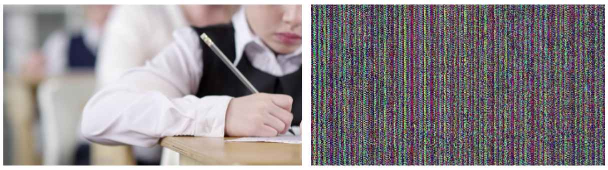

Fig. 3. Original and Encrypted I-Frame.

with 1 to obtain the least significant bit, which is the output of the key selection process.

The second formula encrypts the sign bit of coefficients. It follows all the steps in the first formula.

In both cases, the dimensions of a frame and the current macroblock positions are used as offsets. The encrypted bit is now placed in its original place. To increase efficiency and performance, the Inverse of DCT is not considered on the encryption side but on the decryption side. This approach is known as Transform-Domain encryption.

The decryption process is the same as the encryption, except theis applied IDCT to the decrypted block to retrieve the original video frame. After processing, each frame generates new key sets and key matrices. The encryption algorithm is shown below:

Algorithm 1: Macroblock Encryption

```
Input: Macroblock to be encrypted using the 256-bit
  generated key and key matrix
Output: Encrypted macroblock
Select a key and key matrix for each channel (i.e., Y, U, and V)
  of 16x16 macroblock from a video slice, based on frame
  width and height
Convert the macroblock into float. Then convert the $16 \times 16$
  macroblock into blocks of $8 \times 8$ blocks of each channel
for each $8 \times 8$ block of channel do
    Perform DCT
    Find the 6 DC and AC coefficients in zigzag order, and
      find bits of each coefficient
    for all 11 bits do
        Get the bit by selecting as mentioned earlier the key
          value LSB and key-matrix value LSB
        Perform XOR with the bit of a coefficient and bit from
          the above step
        Replace the bit at the respective index
    end
end
Perform XOR with the sign bit of the coefficient and bit by
  selecting as mentioned earlier the key value LSB and
  key-matrix values LSB
Replace the value at a particular place in a 8x8 block of a
  channel
```

7. RESULTS AND ANALYSIS

This section displays the outcomes of utilizing the produced key to encrypt the 11 bits and the sign bit. As a result, Figure 3 shows the original I-frame of the used video file and the corresponding encrypted frame by changing the DCT values up to 11 bits and sign bit. The same is applicable for the P and B frames. Figure 4 and Figure 5 depict the original and encrypted B-frame and P-frame of the used video file, respectively.

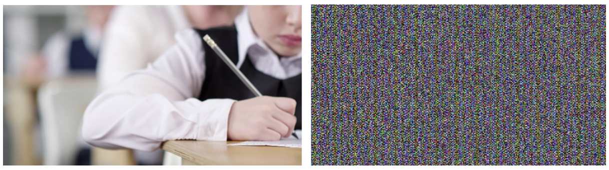

Fig. 4. Original and Encrypted B-Frame.

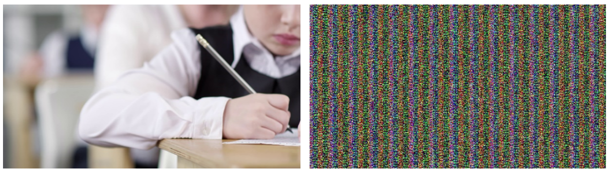

Fig. 5. Original and Encrypted P-Frame.

The above results show the robustness of the algorithm by encrypting each macroblock. Each macroblock of a channel takes ≤ 4 milliseconds to encrypt without adding overhead. For example, each frame has 240×135 macroblocks. To process 30 frames per second, the proposed algorithm uses $240 \times 135 \times 30 \times 72\,(6\,coefficients \times 12\,bits$ [including the sign bit]) bytes per second, which is about 8,748 kbps. While testing the code, it has been noticed that the Intel Core i7 or AMD Ryzen 7 processor and 8 GB of RAM ensure smooth processing of the above data per second.

From the generated 256-bit keys, only $6 \times 12 = 72$ bits is used for encryption and decryption. Assuming a brute force attack to search for the correct combination of 72 bits, an attacker has 184 unused bits from each key. To find a combination of 72 bits, an attacker must select 72 out of the 256 bits. If an attacker wants to search for a specific 72-bit combination through a brute-force search, it will require trying each possible combination until the correct one is found. Assuming the Brute Force attack performs 1 billion i.e.

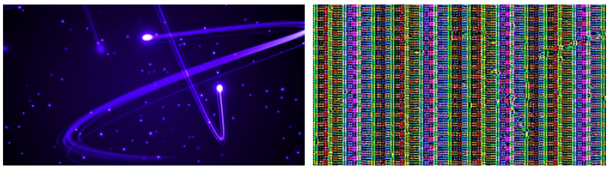

Fig. 6. Original and Encrypted 576p Frame.

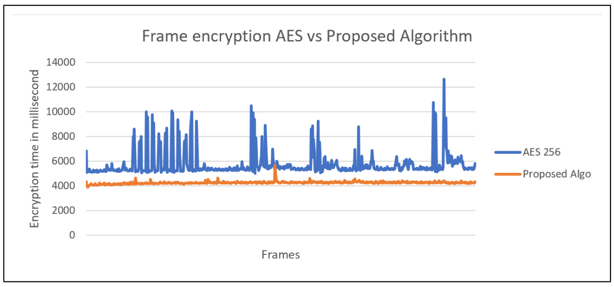

Fig. 7. Comparison of encrypting same frames using AES 256 and Proposed algorithm of 256 bits key.

10^9 attempts per second to find the correct key, the expected time to search through all possible 72-bit combinations would be:

$$\frac{2^{72}}{10^9 \times 60 \times 60 \times 24 \times 365}\, years$$

$$= \frac{4.72 \times 10^{21}}{10^9 \times 60 \times 60 \times 24 \times 365}\, years$$

$$\approx 149670.218\, years$$

Therefore, a brute-force search to find a specific 72-bit combination from a 256-bit key of each is not feasible. The proposed algorithm was also tested on a 576p (a video display resolution) video frame, which has 1024×576 dimensions (Figure 6). It took ≤ 1.2 second to encrypt the whole frame by applying the same algorithm.

There is a possibility that the processing time of each frame in this algorithm could be higher than that of the existing algorithms. It is due to the frame dimensions (that are processed) are different than the other algorithms and lesser macroblocks as well. The lower the dimensions, the less time the algorithm takes to encrypt the frame. Figure 7 depicts the comparison results between AES 256 algorithms with the proposed algorithm with the XOR and 256-bit key to encrypt all the frames. The average time taken by the proposed algorithm is ~ 4.23 seconds to encrypt one frame, whereas the average time taken by AES 256 to encrypt a frame is ~ 5.71 seconds. The result shows that the proposed algorithm performs better than the AES algorithm. In terms of percentage, the proposed algorithm takes 25.92% lesser time than the AES. Earlier results also showed the robustness of the encryption of the proposed algorithm. In order to compare the results with the AES 256 algorithm, the OpenSSL 3.0.7 library is used.

8. CONCLUSION AND FUTURE WORK

In this paper, a selective encryption algorithm is proposed for real-time video encryption. The proposed approach balances encryption strength and video quality. The goal of this work is to encrypt the frame using less computational time than other encryption techniques such as AES, DES, or IDEA encryption algorithms. These algorithms have a high computational cost because of the multiple rounds of encryption. In this specific approach, the XOR operation is applied on 11 bits of coefficient and the sign bit to 6 DCT in the macroblock, using the 256-bit generated keys and key matrices. The XOR takes minimal time to manipulate the bit, and performing the XOR again makes retrieving the same data with the same key possible. Overall, this encryption technique provides robust security for video data. The recovery of original DCT values is possible also by corresponding decryption technique. However, it is essential to note that encryption alone is not enough, and other security measures, such as access controls and authentication, should also be implemented to ensure comprehensive security.

—Future work could potentially harness the benefits of this approach by determining the motion vectors for the P-Frame and B-Frame, paving the way for significant advancements in the video processing algorithms.

—This algorithm might affect the computer performance with the lower configuration, and we are working towards making it efficient for lower configurations in future work.

—We will work on audio channel encryption to encrypt both video and audio data of an MPEG-4 video file.

9. REFERENCES

[1] Encoding process. https://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/jpeg/jpeg/encoder.htm.

[2] Mohamed Abomhara, Omar Zakaria, and Othman Khalifa. An overview of video encryption techniques. *International Journal of Computer Theory and Engineering*, 2(123):103–110, 2010.

[3] Tadetokunbo Abayomi Adenowo and Latifat Folake Oderinu. A comparative study of video cryptographic algorithms and the performance metrics used in the literature to measure the algorithms. *African Journal of Computing and ICT*, 13(4):43–61, December 2020.

[4] Bharat Bhargava, Changgui Shi, and Sheng-Yih Wang. Mpeg video encryption algorithms. *Multimedia Tools and Applications*, 24:57–79, September 2004. https://doi.org/10.1023/B:MTAP.0000033983.62130.00.

[5] Cyril Fonteneau, Jean Motsch, Marie Babel, and Olivier Deforges. A hierarchical selective encryption technique in a scalable image codec. In *International Conference in Communications*, pages 1–4, 2008.

[6] Anish Goel and Kaustubh Chaudhari. Fpga implementation of a novel technique for selective image encryption. In *2nd International Conference on Frontiers of Signal Processing (ICFSP)*, pages 15–19. IEEEXplore, 2016.

[7] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[8] Heinz Hofbauer, Thomas Stütz, and Andreas Uhl. Selective encryption for hierarchical mpeg. *Communications and Multimedia Security. CMS 2006. Lecture Notes in Computer Science*, 4237:151–160, 2006.

[9] Dr. Vikas Saxena Jolly Shah and. Video encryption: A survey. *International Journal of Computer Science Issues*, 8(2):525–533, March 2011.

[10] Lala Krikor, Sami Baba, Thawar Arif, and Zyad Shaaban. Image encryption using dct and stream cipher. *European Journal of Scientific Research*, 32(1):48–58, January 2009.

[11] Zhiyong Li, Xingjun Wang, and Wenming Yang. A fast selective video encryption algorithm by selecting data randomly. In *6th International Conference on Electronics and Information Engineering*. Society of Photo-Optical Instrumentation Engineers (SPIE), 2015.

[12] Gang Liu, Takeshi Ikenaga, Satoshi Goto, and Takaaki Baba. A selective video encryption scheme for mpeg compression standard. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(1):194–202, January 2006.

[13] Rohit Malladar and R. Sanjeev Kunte. Selective video encryption based on entropy measure. *Integrated Intelligent Computing, Communication and Security*, 771:603–612, 2018.

[14] Tameem Obaida, Abeer Salim Jamil, and Nidaa Hassan. A review: Video encryption techniques, advantages and disadvantages. *Webology*, 19(1):7209–7222, March 2022.

[15] Fei Peng, Xiao qing Gong, Min Long, and Xing ming Sun. A selective encryption scheme for protecting h.264/avc video in multimedia social network. *Multimed Tools App*, 76:3235–3253, 2016.

[16] F. Sbiaa, S. Kotel, M. Zeghid, R. Tourki, M. Machhout, and A. Baganne. A selective encryption scheme with multiple security levels for the h.264/avc video coding standard. In *International Conference on Computer and Information Technology (CIT)*, pages 391–398. IEEEXplore, 2016.

[17] G.A. Spanos and T.B. Maples. Performance study of a selective encryption scheme for the security of networked, real-time video. In *Proceedings of Fourth International Conference on Computer Communications and Networks*, pages 2–10. IEEEXplore, 1995.

[18] N. M. Thomas, D. Lefol, D. R. Bull, and D. Redmill. A novel secure h.264 transcoder using selective encryption. In *IEEE International Conference on Image Processing*, pages 85–88. IEEEXplore, 2007.

[19] Qiuhua Wang and Xingjun Wang. A new selective video encryption algorithm for the h.264 standard. In *International Conference on Progress in Informatics and Computing*, pages 275–279. IEEEXplore, 2014.

[20] Yajun Wang, Mian Cai, and Feng Tang. Design of a new selective video encryption scheme based on h.264. In *International Conference on Computational Intelligence and Security (CIS 2007)*. IEEEXplore, 2007.