Efficient Video Processing Schemes for Compression and Security

Thesis submitted by

Jagannath Sethi

Doctor of Philosophy(Ph.D.)
in
Engineering

Dept. of Electronics &
Telecommunication Engineering,
Faculty Council of Engineering & Technology
Jadavpur University, Kolkata
India

2026

JADAVPUR UNIVERSITY
KOLKATA-700 032, INDIA

INDEX NO.: 250/20/E

1. **Title of the Thesis:**

   "Efficient Video Processing Schemes for Compression and Security"

2. **Name, Designation & Institution of the Supervisor:**

   Prof. Ananda Shankar Chowdhury

   Professor

   Department of Electronics & Telecommunication Engineering

   Jadavpur University, Kolkata, India

3. **Name, Designation & Institution of the Co-Supervisor:**

   Prof. Jaydeb Bhaumik

   Professor

   Department of Electronics & Telecommunication Engineering

   Jadavpur University, Kolkata, India

   List of publications:

   (Journal Published)

   1. **J. Sethi**, J Bhaumik, A.S. Chowdhury: "Joint Video Compression and Encryption using Parallel Compressive Sensing and Improved Chaotic Maps", *Journal of Digit. Signal Process., Elsevier*, 130, p. 103746, 2022.
   2. **J. Sethi**, J Bhaumik, A.S. Chowdhury: "A Robust and Secure Video Recovery Scheme with Deep Compressive Sensing" *Image Vis. Comput., Elsevier*, 166, p. 105853, 2026.

   (Journals Communicated)

1. **J. Sethi**, J Bhaumik, A.S. Chowdhury: "A Unified Video Compression and Encryption Scheme for WVSNs", *In preparation.*

(In International Conference)

1. **J. Sethi**, J. Bhaumik, A.S. Chowdhury: "Chaos-Based Uncompressed Frame Level Video Encryption", *Seventh International Conference on Mathematics and Computing (ICMC), D. Giri et al. (eds.), Singapore Advances in Intelligent Systems and Computing, vol 1412. Springer, Singapore. (2021), pp. 201 -217.*
2. **J. Sethi**, J. Bhaumik, A.S. Chowdhury: "Fast and Secure Video Encryption Using Divide-and-Conquer and Logistic Tent Infinite Collapse Chaotic Map", *Sixth International Conference on Computer Vision and Image Processing (CVIP), B. Raman et al. (eds.), Springer Communications in Computer and Information Science, vol 1567, Ropar, India (2021), 151-163.*
3. **J. Sethi**, M. Mondal, J. Bhaumik, A.S. Chowdhury: "Deep Compressive Sensing for High-Quality Video Recovery", *Ninth International Conference on Computer Vision and Image Processing (CVIP), Kancheepuram, India (2024), 119-134.*

4. List of Patents: NIL

5. List of Presentations in National/International Conferences:

1. Presented a paper entitled "Chaos-Based Uncompressed Frame Level Video Encryption", *Seventh International Conference on Mathematics and Computing (ICMC), 02-05 March, 2021, Singapore, 201 -217.*
2. Presented a paper entitled "Fast and Secure Video Encryption Using Divide-and-Conquer and Logistic Tent Infinite Collapse Chaotic Map", *Sixth International Conference on Computer Vision and Image Processing (CVIP), 03-05 December, 2021, Ropar, India.*

3. Presented a paper entitled "Deep Compressive Sensing for High-Quality Video Recovery", *Ninth International Conference on Computer Vision and Image Processing (CVIP), 19-21 December, 2024, Kancheepuram, India.*

STATEMENT OF ORIGINALITY

I, ***Jagannath Sethi***, registered on ***20/02/2019*** do hereby declare that this thesis entitled "***Efficient Video Processing Schemes for Compression and Security***" contains a literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All the information in this thesis has been obtained and presented following existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the Policy on Anti Plagiarism, Jadavpur University, 2019, and the level of similarity as checked by iThenticate software is 6%.

Jagannath Sethi

Signature of the Candidate:

Date: 08/04/2026

Certified by Supervisors:

Signature of the Supervisor:

Prof. Ananda Shankar Chowdhury

Professor, Department of ETCE

Jadavpur University, Kolkata, India

ANANDA SHANKAR CHOWDHURY
Professor
Electronics and Telecommunication Engineering
Jadavpur University, Kolkata-32

Jaydeb Bhaumik

Signature of the Co-Supervisor:

Prof. Jaydeb Bhaumik

Professor, Department of ETCE

Jadavpur University, Kolkata, India

Dr. Jaydeb Bhaumik
Professor
Dept. of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled "***Efficient Video Processing Schemes for Compression and Security***" submitted by ***Shri Jagannath Sethi***, who got his name registered on ***19.02.2020*** for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of ***Prof. Ananda Shankar Chowdhury*** and ***Prof. Jaydeb Bhaumik*** and that neither his thesis nor any part of this thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Signature of the Supervisor

Dr. Ananda Shankar Chowdhury

Professor, Dept. of ETCE

Jadavpur University, Kolkata – 700032

Date: 08/04/2026

ANANDA SHANKAR CHOWDHURY
Professor
Electronics and Telecommunication Engineering
Jadavpur University, Kolkata-32

Signature of the Co-Supervisor

Dr. Jaydeb Bhaumik

Professor, Dept. of ETCE

Jadavpur University, Kolkata – 700032

Date: 08/04/2026

Dr. Jaydeb Bhaumik
Professor
Dept. of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032

Abstract

This thesis addresses the important challenges of video encryption, compression, and high-quality recovery through a series of novel methods based on chaos theory and deep learning. We develop robust, compression-independent video encryption systems operating at both the frame and shot levels in the first set of contributions. Using a logistic sine coupling map to create a pseudorandom chaotic sequence for block-level diffusion, a frame-level strategy presents a novel confusion mechanism based on sort index permutation and thus improves robustness against known and chosen plaintext attacks. Complementarily, a shot-level approach exploits temporal redundancy to derive sparse difference frames, and it uses a newly proposed logistic tent infinite collapse map (LT-ICM) to both permute and substitute pixel values effectively, ensuring rapid encryption with enhanced security and efficiency. Building on this foundation, the thesis also investigates video Compressive Sensing (CS) techniques based on deep learning that are intended for use in cloud-based surveillance and telemedicine. To optimize sampling process and recover high-quality video data, videos are divided into Groups of Pictures (GOPs) that comprise both keyframes and non-keyframes. A convolutional neural network (CNN) with a loss function based on the Structural Similarity Index Measure (SSIM) is then used. The recovery process utilises an initial spatial reconstruction stage followed by a deep recovery stage that implements multilevel and single-level feature compensation to effectively restore non-keyframes using contextual information from both keyframes and neighboring non-keyframes. A joint compression and encryption approach is developed, using Parallel Compressive Sensing (PCS) in combination with improved chaotic maps. This method analyses video in GOP segments, applying DCT-based sparsification to reference frames and then permuting coefficients, while the measurement matrix is produced using a one-dimensional LT-ICM. We introduce a novel substitution method that utilises circular shift, XOR, and modular addition operations. We finally shuffle the substituted frames using a frame-dependent two-dimensional

LT-ICM chaotic sequence. Experimental analysis, including comparisons with state-of-the-art approaches, establishes the effectiveness of the proposed solution. We propose a secure, high-quality video recovery framework that seamlessly integrates deep learning-based compressed sensing (CS) with an advanced encryption strategy. This scheme not only capitalises on the superior recovery capabilities of the CNN-based two-stage process but also reinforces data security through a multi-phase encryption routine (forward diffusion, substitution, backward diffusion, and XORing) powered by an unpredictably complex Sine Symbolic Chaotic Map (SSCM). Comprehensive experimental analyses across all methods demonstrate significant improvements in encryption speed, correlation robustness, entropy measures, and overall reconstruction quality, thereby outperforming several state-of-the-art techniques. Finally, an adaptive shot-level unified framework is proposed by integrating adaptive parallel compressive sensing (APCS) with a secure encryption pipeline. In this framework, shot boundaries are first detected, followed by adaptive sampling within each shot. A highly unpredictable one-dimensional chaotic map Log–Logistic–Cubic (LLC) is used for both measurement matrix generation and encryption operations, while the matrices are further optimized through QR decomposition with Givens rotations to enhance reconstruction quality. Overall, the proposed framework achieves an effective balance between compression efficiency, low-sampling-ratio recovery, and video security. Collectively, the contributions of this thesis establish innovative paradigms for secure and efficient video processing, paving the way for reliable applications in environments demanding high security.

Acknowledgment

"The purpose of life is not to be happy. It is to be useful, to be honorable, to be compassionate, to have it make some difference that you have lived and lived well".

— Ralph Waldo Emerson

The fulfillment derived from the successful attainment of a PhD is incomplete without acknowledging those individuals whose unwavering direction and support have culminated in this achievement.

There is no proper way to acknowledge everyone whose consistent effort, assistance and guidance have played a key role in this journey and finally leading to this dissertation. First and foremost, I would like to offer my profound gratitude to my advisor Prof. (Dr.) Ananda Shankar Chowdhury and Prof. (Dr.) Jaydeb Bhaumik, for their invaluable guidance, counsel throughout my research work, and meticulous correcting my papers. They were very helpful with key suggestions while solving different problems addressed in the thesis. I was fully allowed to have the necessary freedom to exercise thoughtful and scientific approaches to the problem and have free discussions on them. He has not only helped me in bringing the thesis to this shape, but also stretched his helping hands whenever I was in need. I consider myself fortunate to have them as my advisors.

I would also like to express my heartfelt gratitude to Prof. Amit Konar, Prof. Mrinal Kanti Naskar, Prof. Sanjoy Kumar Saha, Prof. Sudhabindu Ray, Prof. P. Venkateswaran, and Prof. Selhi Sinha Chowdhury for their insightful suggestions in various aspects of this wonderful journey.

I want to extend my gratitude to the following wonderful members of Imaging, Vision and Pattern Recognition (IVPR) group, who were helpful at various instants: Abhimanyu Sahu, Arindam Sikdar, Rukmini Roy, Arijit De, Sushanta Sahu, Sevakram T. Kumbhare and Anirban Dutta Choudhury. I would also like to thank Moinak Mondal, who also helped with some of the problems related to this dissertation during his Master course. Without their help, this work would be much difficult to complete.

Last but not the least, I would like to express my heartfelt regards to my Brothers, Balaram Sethi, Sudarsan Sethi, Gadadhar Sethi and sister Mamata Sethi for their lifelong guidance, support and faith that always motivates me to work harder and better. I am thankful to my beloved life partner Deeptimayee Sethy and my sweet daughters, Krishita and Harshita, who believed in me and supported me during the entire course of the research study. Without the support of my wife, I would not have completed my Ph.D. work. I also thank my mother-in-law, Mrs. Namita Sethy, and my father-in-law, Mr. Mahendra Sethy, for providing support and believing in me throughout the course of my research work.

Needless to say, without all the above help and support the writing and production this thesis would not have been possible.

Jagannath Sethi

Jagannath Sethi

Jadavpur University

Kolkata – 700032

Date: 08/04/2026

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The thesis work focuses on different efficient video encryption, compression, and joint video compression and encryption schemes. More particularly, we investigate video encryption of uncompressed video using chaotic maps, video compression using deep compressive sensing, and joint video compression and encryption using compressive sensing and chaotic maps. Section 1.1 addresses the motivation for the thesis work. In section 1.2, we discuss video encryption schemes. Section 1.3 provides an overview of video compression, considering the difference between traditional and deep learning-based approaches. Section 1.4 describes the joint video compression and encryption methods. We examine the research gaps in section 1.5. Section 1.6 summarises the primary contributions. Section 1.7 summarises the organisation of the remainder of this thesis.

1.1 Motivation

Efficient video procession concerning encryption and compression is an important topic for real time application and resource constraint environment [1, 2]. Video signals, which are used in applications such as conferencing, surveillance, and social media, need a substantial quantity of data to be sent across channels that have a fixed bandwidth. These challenges give rise to concerns over the limited bandwidth for transmission, the costs of storage, and the safety of the data. While robust encryption ensures secure transmission across open or unprotected networks, video

compression helps to alleviate problems with bandwidth and storage space [2]. Joint video compression and encryption techniques are an effective way to compress and secure videos at the same time, which are suitable for resource-constrained applications [2–4].

The motivation behind this thesis is to address issues related to video encryption, video compression, and joint video compression and encryption, all of which are suitable for applications with limited resources.

1.2 Video Encryption

Video encryption has become an essential need for protecting the privacy and authenticity of visual data due to the meteoric rise of digital video content and its extensive distribution across public networks [5]. The Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two conventional encryption schemes that have been extensively employed to secure multimedia content. Nevertheless, these methods are frequently hampered by their high computational complexity and inefficiency in managing the redundancy and large data size of video streams, rendering them unsuitable for real-time applications [6]. Researchers have used more specialized techniques, such as chaos-based encryption, to address these challenges. In some applications, this kind of encryption may provide superior security and perhaps lower processing costs [7].

Video encryption is the technique by which a user encrypts a video using a key such that only allowed users with the key can decrypt it, and no unauthorized users can since the key is only known to authorized users. In Fig.1.1, we illustrate the video encryption and decryption method. Video encryption techniques using chaotic maps have gained considerable attention as an alternative method due to their fundamental attributes, including sensitivity to initial conditions, controlling parameters, ergodicity, and pseudo-randomness. These properties enable chaotic maps to be highly efficient in cryptographic applications, offering robust security while maintaining low computational complexity [8, 9].

Video encryption techniques may be categorized into two approaches: (a)

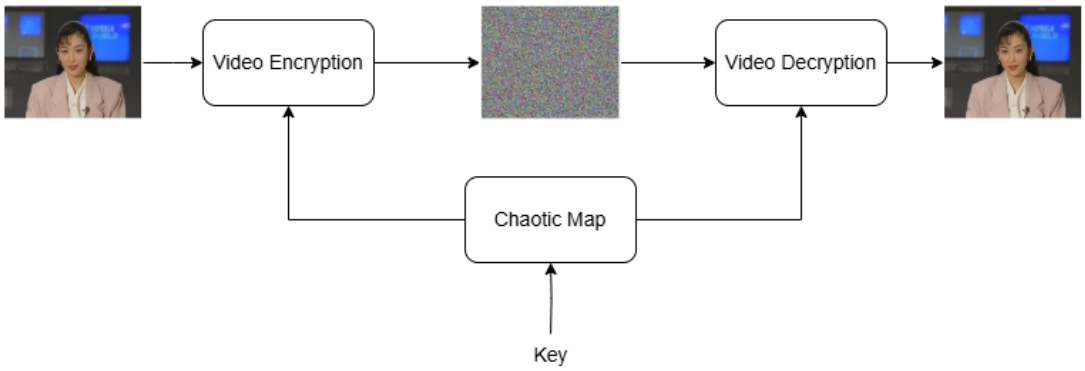


Figure 1.1: Block diagram of video Encryption and decryption

compression-dependent approach and (b) compression-independent approach. Encrypting uncompressed video is vital in cases where maintaining high quality is essential, such as in medical imaging, military applications, or safeguarding sensitive video collections. This approach ensures immediate security against unauthorized access, independent of compression methods. Chaotic encryption is particularly advantageous due to its high security and computational efficiency, suitable for protecting uncompressed video.

In multimedia scenarios, encrypting compressed video is essential due to the prevalence of compressed content for bandwidth and storage efficiency. Encrypting compressed videos offers increased efficiency and reduced computational load since fewer samples or compressed formats require encryption. Chaotic-map-based encryption for compressed video data, utilizing either conventional coding techniques [10] or deep learning (DL) methods, is appealing as it ensures secure encryption with low computational requirements, aligning well with existing compression processes. So there is a need to study both compression-independent and dependent video encryption.

In general, confusion and diffusion are two important properties in any cryptographic scheme. In confusion, the pseudorandom sequence alters the position of the data, while in diffusion, it modifies the values of the confused data [9]. The pseudorandom sequence is generated using the chaotic map. The confusion and diffusion-based video encryption scheme is shown in Fig.1.2.

Chaotic maps are essentially an iterative nonlinear function, as shown in Fig.

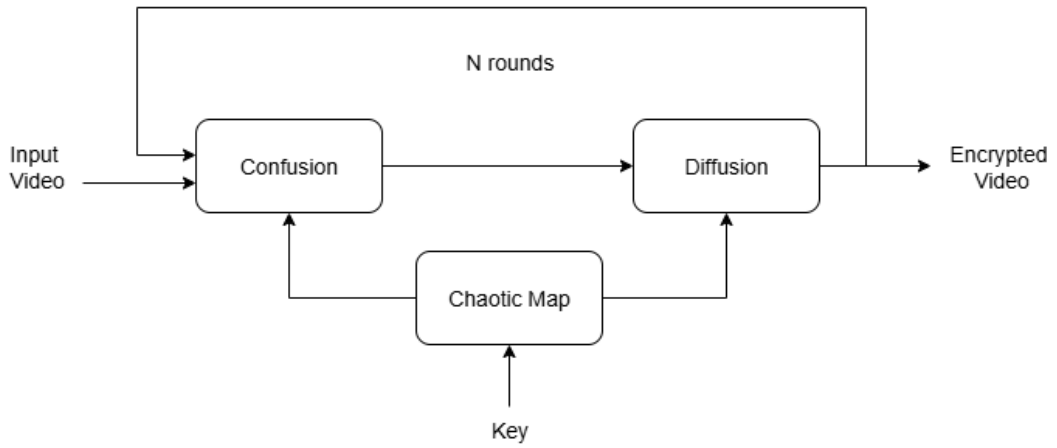


Figure 1.2: Video Encryption based on confusion and diffusion

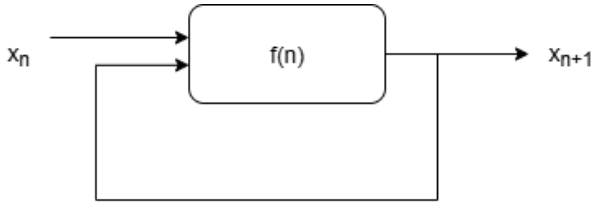
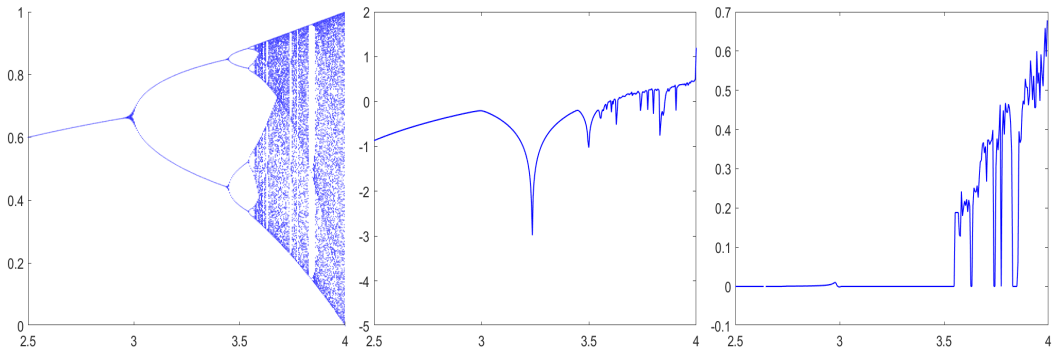


Figure 1.3: Chaotic map as iterative function

1.3, which are highly sensitive to initial values and control parameters, resulting in complex chaotic properties. The different chaotic maps are discussed in [4, 11–13]. The chaotic maps are evaluated using Lyapunov exponents (LE), sample entropy (SE), and bifurcation diagrams. High positive values for LE and SE are necessary for a very unpredictable chaotic map which can be used for video encryption [14].

The most often used chaotic maps for video encryption are Henon, Tent, Sine, and Logistics. These maps all have a small range of chaos. The bifurcation diagram, LE and SE for the Logistic map, is shown in Fig.1.4. The chaotic range for the Logistic map is between 0.57 and 4. In order to make the pseudorandom sequence hard to predict, a chaotic map should have a wide chaotic range. We propose several chaotic maps with a higher chaotic range, LE and SE. These chaotic maps are suitable for high-security video encryption.



(a) (b) (c)

Figure 1.4: Analysis of Logistic map: (a) Bifurcation diagram, (b) Lyapunov exponent, (c) Sample entropy

1.3 Video Compression

Video compression is a method to compress video data so that it can be represented using fewer bits, needs low transmission bandwidth, and requires low memory for storage. Data can be compressed in two ways. Firstly, it can be compressed using a traditional coder, which deals with sampling, quantizing, and encoding. In general, the sampling theorem has to be used for decompression in traditional data compression methods. A video can be compressed using traditional methods using video codecs like H.264, H265, and VVC which deal with motion compensation, transformation, and encoding. Video compression employing various codecs requires intricate encoders that demand substantial computational power for processing video data, in contrast to decoders, making such methods inappropriate for low-power applications [4].

Even though these codecs are good for compressing data, they have problems like having low latency and not being able to adapt to new multimedia situations like the Internet of Things or low-power sensor networks. This is why researchers are looking into other ways to compress video data.

A typical example of video compression and decompression is illustrated in Fig.1.5. A new method of video compression based on Compressive Sensing (CS) has evolved that takes far fewer measurements than those required by the Nyquist-Shannon sampling theorem [15]. Reconstructing video with CS requires much fewer

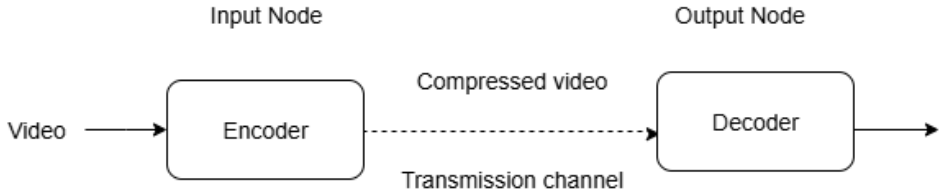


Figure 1.5: Video encoding and decoding

samples than conventional sampling methods because it makes use of signal sparsity [16]. Compressed sensing enables the capture and recovery of signals or images at a sub-Nyquist-Shannon sampling rate [17]. Consequently, CS has drawn significant interest and has been extensively used in several imaging domains, including magnetic resonance imaging (MRI) [18], snapshot compressed imaging [19], and video compressing sensing [20].

Two fundamental concerns in CS applications are the measurement matrix design and the reconstruction approach [21, 22]. The sampled data can be recovered using the measurement matrix [16]. Data sampling can be achieved using Eq.1.1.

$$y = \Phi \times x \tag{1.1}$$

where x, y, and Φ denote input data, and sampled data and the measurement matrix, respectively, with dimensions $N \times 1$, $M \times 1$, and $M \times N$.

Measurement matrices can be designed using Gaussian, Bernoulli, and Toeplitz matrices. When the video is compressed at lower sampling rates, like 0.01 and 0.05, the measurement matrix that is made from these matrices can't provide good recovery quality because there is very little information that holds the signal structure, even if the signal is sparse. Since there is extremely little chance that the matrix would preserve distances across all sparse vectors, it is challenging to meet the Restricted Isometry Property (RIP) [23]. During the recovery process at lower sample ratios, these matrices do not take into account other information such as the temporal and spatial structure of the video. However, these measurement matrices can give good results at high sampling ratios like 0.5, 0.6, etc.

The chaotic map can also be employed to generate the measurement matrix,

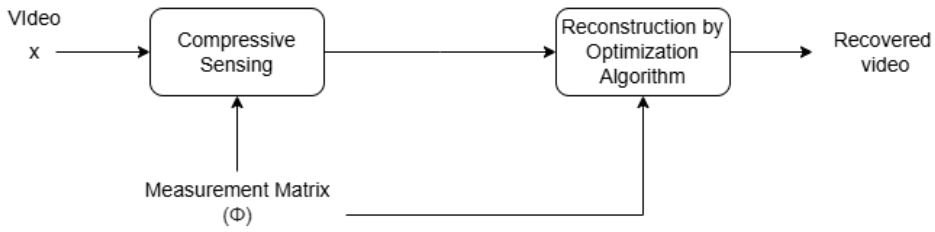


Figure 1.6: Video compression and decompression using CS

which demonstrates superior recovery [4]. Fig.1.6 illustrates the process of video compression and decompression using a measurement matrix.

Different multiple hypothesis (MH)-based methods ([24], MH-ST [25], MH-RTIK [26]) are used in the recovery video. However, these methods are not suitable for very low sampling ratios like 0.01 and 0.05 because it is extremely difficult to recover high-quality video with a handful of samples. The major drawbacks of MH methods are high computational complexity and sensitivity to the hypothesis. The sampled data can be recovered with good quality using different recovery methods, considering y as linear projections from x under the ill-condition matrix Φ, and the recovery of x is an underdetermined inverse problem. Orthogonal matching pursuit (OMP) [27] and subspace pursuit (SP) [28] are greedy algorithms that have been used in CS to tackle this issue. These algorithms aim to properly search for the atoms involved in the measurement process; nevertheless, greedy algorithms may get trapped in local optima, limiting reconstruction performance.

The traditional reconstruction techniques such as orthogonal matching pursuit (OMP) and basis pursuit are impacted by signal sparsity, measurement matrix size, and noise sensitivity. The generation of measurement matrix is content independent. So during the recovery scheme, it can not provide good quality video. So these methods can not give better recovery quality at low sampling ratios like 0.01 and 0.05. However, deep learning-based CVS where the measurement matrix is optimized using the convolution layer as opposed to a content-independent measurement matrix. The DL-based method can sample the data at low sampling ratios (like 0.01 and 0.05) with good recovery quality.

Recent deep learning-based image CS techniques [29–31] may be used to achieve

high-quality recovery. However, these image-based CS methods used to compress video always omit temporal information between frames, which makes the recovered video poor quality. To enhance video quality, Shi et al. [20] presented an end-to-end DL-based video CS scheme called video compressive sensing network (VCSNet), which uses convolutional neural networks (CNN). They use keyframes that are sampled at a higher rate to compensate for non-keyframes, which are sampled at a lower rate. They used a mean square error (MSE)-based loss function, which performs much worse than our structural similarity index measure (SSIM)-based loss function.

1.4 Joint Video Compression and Encryption using Compressive Sensing

Video encryption methods [32, 33] encrypt the video to prevent any authorized user from recognizing it without a key. However, the sizes of both the original video and the encrypted video remain the same for these methods. The video data can be sampled and compressed using compressive sensing. A few CS-based cryptosystem methods are described in [34, 35]. In CS, the data can be compressed using a linear operation between a measurement matrix Φ and the original data (x). In general, random matrices like Gaussian and Bernoulli are used to generate measurement matrices, which cannot provide better security on their own. So there is a need for joint video compression and encryption, or making the compression and encryption in an integrated manner to achieve simultaneous compression and encryption. For simultaneous compression and encryption of video data, compressive sensing-based techniques are used where the video is both sampled and encrypted. Video must first be compressed using CS to eliminate redundancy. After compression, encryption will be performed in order to randomise the compressed video [36].

Video compression focuses on reducing redundancy, while encryption relies on randomizing it. Achieving balance in a hybrid compression and encryption strategy is challenging because it must retain high compression efficiency while providing improved security. The design of the measurement matrix is also a challenging issue

concerning its reproducibility and complexity. Gaussian and Bernoulli matrices, due to their higher computational complexity and storage requirements, may not be suitable for resource-constrained applications where both video compression and encryption are required. For video data, it is crucial to handle parallel processing when considering compression and encryption in an integrated manner. The choice between the traditional and DL-based CS in joint compression and encryption is a crucial thing to consider with respect to recovery quality, speed, robustness, and resources.

In this thesis, we have focused on employing chaotic maps to encrypt both compressed and uncompressed video for better security. Furthermore, we have concentrated on developing DL-based video CS techniques to increase compression efficiency while maintaining high video frame recovery quality. Also, taking into account both traditional and deep learning-based compressive video sensing, we have focused on integrating compression and encryption to attain overall efficiency in both processes.

1.5 Research Gap

Although significant progress has been made in video encryption, video compression, and joint video compression-encryption, several important challenges still remain unresolved. Existing video encryption methods often suffer from either high computational complexity, limited diffusion capability, or inadequate exploitation of temporal redundancy among consecutive frames. Many conventional and chaotic-map-based schemes process frames independently, which restricts the propagation of plaintext changes across the video sequence and limits robustness. In addition, several existing chaotic maps used for encryption have a narrow chaotic range or reduced unpredictability, which may weaken security performance.

Similarly, existing video compression methods based on conventional codecs require computationally intensive encoders, making them unsuitable for low-power and resource-constrained platforms. Compressive sensing (CS) offers a promising alternative because of its low encoder complexity; however, traditional CS reconstruc-

tion methods often fail to recover high-quality video at very low sampling ratios. Measurement matrices generated from conventional random models are content-independent and do not effectively preserve the structural and temporal characteristics of video data during recovery. Although deep learning-based compressive sensing methods improve reconstruction quality, many image-based schemes neglect temporal correlation when applied to video, thereby limiting their effectiveness.

The problem becomes even more challenging when compression and encryption must be performed jointly. In such cases, the system must achieve both high compression efficiency and strong security while remaining computationally feasible for resource-constrained applications such as wireless video sensor networks. Existing joint schemes are often based on fixed GOP structures, non-adaptive sampling strategies, or multi-round encryption mechanisms, which either reduce recovery quality at low sampling ratios or increase computational overhead. Therefore, there is a clear need for new methods that can exploit temporal redundancy more effectively, support adaptive sampling, improve low-sampling-ratio recovery, and provide strong security with reduced complexity.

Motivated by these observations, this thesis develops a sequence of methods that progressively address these research gaps through improved chaotic-map-based video encryption, deep learning-based video compressive sensing, and efficient unified compression–encryption frameworks.

1.6 Contributions of the Thesis

The contributions of this thesis have been developed in a progressive manner, where each subsequent work addresses specific limitations observed in the preceding approaches. The research begins with secure and computationally efficient encryption schemes for uncompressed video, then advances toward deep learning-based video compressive sensing for improved reconstruction at low sampling ratios, and finally culminates in unified frameworks that jointly perform video compression and encryption. Accordingly, the contributions of this thesis are organized into three subsections: video encryption, video compression, and unified video compression and

encryption.

1.6.1 Contributions in Video Encryption

The first set of contributions of this thesis focuses on efficient and secure video encryption of uncompressed video. Initially, a compression-independent video encryption scheme is proposed in which video frames are processed using sort-index-based permutation and block-level diffusion. In this method, the chaotic sequence used for permutation and substitution is made plaintext-dependent, thereby improving security against brute-force, known-plaintext, and chosen-plaintext attacks.

A second contribution in this direction is a divide-and-conquer encryption framework for uncompressed video. This method exploits temporal redundancy within each shot through difference-frame processing so that only a limited number of significant non-zero values need to be handled during encryption. As a result, the proposed method improves computational efficiency while maintaining strong security. In addition, improved chaotic maps such as LT-ICM are developed and utilized to enhance chaotic range, unpredictability, and sensitivity to initial conditions and control parameters, making them more suitable for robust multimedia encryption.

1.6.2 Contributions in Video Compression

The second set of contributions addresses video compression using deep compressive sensing. In this part of the thesis, deep learning-based video compressive sensing schemes are proposed to achieve high-quality recovery at low sampling ratios. Two recovery networks are developed: a single keyframe-based framework and a double keyframe-based framework. In these methods, keyframes and neighbouring non-keyframes are jointly utilized to compensate for low-sampled non-keyframes, thereby improving the reconstruction quality.

To further improve the recovery quality of video, an SSIM-based loss function is introduced instead of the conventional MSE-based objective. The proposed deep recovery frameworks provide good-quality reconstructed video while maintaining reasonable recovery time, even at very low sampling ratios. These contributions

address the limitations of traditional CS recovery techniques and image-based deep CS methods that do not effectively exploit temporal information in video sequences.

1.6.3 Contributions in Unified Video Compression and Encryption

The final and most significant set of contributions of this thesis lies in unified video compression and encryption. First, a GOP-level joint video compression and encryption framework is proposed using parallel compressive sensing and an improved chaotic map. In this scheme, the chaotic map is used to generate the measurement matrix and provide first-level security during sampling. Additional security is achieved through a new substitution mechanism and data-dependent shuffling operations.

Next, a unified framework combining deep compressive sensing and efficient classical encryption is developed. In this method, video compression is performed using deep CS to ensure high-quality recovery at low sampling ratios, while encryption is carried out using a lightweight bidirectional diffusion mechanism to reduce computational overhead. A new chaotic map with broader chaotic range is also introduced to improve the security of compressed video encryption.

Finally, an adaptive shot-level unified framework is proposed based on adaptive parallel compressive sensing (APCS) and a secure encryption pipeline. In this approach, shot boundaries are first detected, after which adaptive sampling is performed within each shot. A highly unpredictable one-dimensional chaotic map is employed for both measurement matrix initialization and encryption-related operations. The measurement matrices are further refined using QR decomposition with Givens rotations to improve reconstruction quality. This final framework provides an effective balance among compression efficiency, low-sampling-ratio recovery, and strong video security.

1.7 Thesis Organization

The thesis is organized in accordance with the progression of the research work, beginning with video encryption, followed by video compression, and finally advancing toward unified video compression and encryption frameworks. The remaining thesis is divided into the following six chapters.

Chapter 2 This chapter presents two video encryption schemes for uncompressed video. The first scheme employs sort-index-based permutation and block-level diffusion. The second scheme introduces a shot-level video encryption framework based on the proposed chaotic map. In both methods, chaotic maps are utilized to generate pseudorandom sequences. The proposed chaotic map is termed the Logistic Tent Infinite Collapse Map (LT-ICM).

Chapter 3 This chapter presents a convolutional neural network (CNN)-based deep compressive sensing scheme for video. An SSIM-based loss function is introduced to improve reconstruction quality. The proposed framework provides high-quality recovery at low sampling ratios. During the recovery phase, multilevel feature compensation is employed to enhance the quality of the reconstructed video.

Chapter 4 This chapter develops a joint video compression and encryption scheme using parallel compressive sensing and an improved chaotic map. In this work, the one-dimensional LT-ICM is extended to a two-dimensional LT-ICM to achieve better unpredictability. The generated chaotic sequences are made plaintext-dependent to strengthen security. In addition, the sparse data are arranged in ascending order column-wise to improve recovery quality. The proposed substitution mechanism incorporates XOR, circular shift, and modular addition operations to enhance security further.

Chapter 5 This chapter presents a robust and secure joint video compression and encryption scheme at the Group of Pictures (GOP) level. In this framework, compression is achieved using deep compressive sensing, while encryption is performed using an efficient classical encryption approach. In addition, a chaotic map termed the Sine Symbolic Chaotic Map (SSCM), having a broader chaotic range, is proposed. For the encryption of compressed features, a secure scheme consisting of

forward diffusion, substitution, backward diffusion, and XOR operation with an SSCM-based chaotic sequence is developed.

Chapter 6 This chapter proposes a unified framework that combines adaptive parallel compressive sensing (APCS) for video compression with secure encryption. A one-dimensional Log–Logistic–Cubic (LLC) chaotic map is used to generate the measurement matrices for APCS, and these matrices are further refined using QR decomposition with Givens rotations to improve recovery quality. The same LLC-based sequences are also employed for permutation and key-based substitution during encryption, thereby ensuring both computational efficiency and strong security.

Chapter 7 This chapter concludes the thesis and summarizes the major findings of the research. It also outlines possible directions for future work.

Chapter 2

Encryption of Uncompressed Video

In this chapter, we address two uncompressed video encryption schemes using a chaotic map. In our first work, we introduce a method for video encryption at the frame level that operates independently of compression. This method introduces a novel confusion technique based on the permutation dealing with sorting indices. During shuffling, a new sequence is generated for every video frame under consideration. A pseudorandom chaotic sequence is influenced by initial states, controlling parameters, and the video frames. We have employed a logistic sine coupling map (LSCM) to produce sequences with a high degree of randomness. Block-level diffusion techniques have been utilized to obfuscate the frame sequences. The newly designed framework demonstrates enhanced robustness to known and chosen plaintext attacks because it utilizes the total size and addition of pixel values to produce a random chaotic sequence. In second work, we propose a fast and secure method for compression-independent video encryption. A divide-and-conquer type approach is taken, where frames in a video are processed at the shot level. Temporal redundancy among the frames in a shot is exploited next to derive the difference frames, where some of the pixel values are already zero. A further sparse representation of the pixels is obtained by thresholding the non-zero values. We also propose a new chaotic map, namely, the Logistic Tent Infinite Collapse map (LT-ICM), which has better chaotic range, complex behaviour, unpredictability, ergodicity, and sensitivity

towards initial values and controlling parameters. We permute the pixels of each shot (left after thresholding) with reference to the indices of sorted chaotic sequence generated by LT-ICM. Substitution operation is applied next to change the pixel values. The substituted pixels are arranged into frames and pixels are further substituted at frame level with another chaotic sequence generated by LT-ICM. Experimental results establish that our proposed two methods are, efficient in terms of encryption time, correlation coefficients and entropy values compared to some of the existing methods.

2.1 Frame Level Video Encryption using chaotic map

2.1.1 Introduction

The security of multimedia content using chaos theory is a critical area of research aimed at securing images and videos from unauthorized access and security breaches. Ensuring both confidentiality and integrity during data transmission and storage is essential. Applications such as medical imaging, video conferencing, and military imaging systems demand robust security for video data protection during storage and transmission. Traditional encryption methods like the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and International Data Encryption Algorithm (IDEA) are inadequate for real-time digital video encryption [37]. Chaotic systems are favored for video encryption due to their complex chaotic dynamics, sensitivity to initial conditions and control parameters, and properties like ergodicity and unpredictability. Our video encryption method, which operates independently of compression, is suitable for military and defense settings where critical video data is stored long-term, providing attackers with considerable time to conduct brute-force attacks. Our proposed method offers enhanced security due to its expanded key space.

Due to the necessity for a joint compression and encryption algorithm to address multiple concerns such as format and codec compliance, it proves inadequate for

a broad range of applications [32, 38]. One solution for this is to use compression independent video encryption. Several analyses indicate that for permutation use of only cipher is not a secure strategy as this is vulnerable to known and chosen plaintext attacks [39].Therefore, incorporating both confusion and diffusion processes is essential for enhancing security. In the proposed approach, we've utilized confusion along with block-level diffusion to encrypt the uncompressed video.

Our suggested video encryption system is structured around three pivotal stages. Initially, chaotic sequences are produced via the logistic sine coupling map (LSCM) [11]. Next, each frame undergoes permutation based on a sort index to shuffle the frames. A subsequent chaotic sequence is derived from these shuffled frames. Lastly, we conduct block-level diffusion on the shuffled frames to achieve encrypted video frames. The primary innovations of our proposed video encryption algorithm, which operates independently of compression, are as follows:

1. We present a permutation process based on a sort index for the confusion phase. Each frame undergoes shuffling through its unique chaotic sequence. These sequences are refreshed using the preceding shuffled frame. Specifically, when shuffling any frame, the prior shuffled frame alongside the pre-existing chaotic sequence guides the process.

2. We introduce a block-level diffusion technique where a distinct sequence is produced for every block. To generate this sequence for each block, the preceding encrypted block is utilized to update the initial values and control parameters.

The rest of this work is organized as follows. In section 2.1.2, we discuss the related works. In section 2.1.3, we describe our proposed framework, pseudorandom sequence generation, confusion, and diffusion process. In section 2.1.4, we present our experimental results with detailed analysis. Finally, the work is discussed in section 2.1.5.

2.1.2 Related Work

Encryption techniques for video content can be categorized into two distinct methodologies: (a) those reliant on compression, and (b) those independent of compression. The encryption strategies that depend on compression typically do not offer an optimal balance between computational efficiency and the level of security provided, as noted in previous studies [40]. For highly sensitive video applications compression independent methods are preferred for encrypting the video. Various algorithms for encrypting uncompressed video, independent of compression techniques, have been analyzed in [40–43]. Zhang et al. [44] introduced a surveillance video encryption strategy utilizing layered cellular automata that targets only encrypting regions of interest. Additionally, some techniques [45–47] focus on partial video encryption to enhance encryption speed. Nonetheless, these selective encryption strategies fall short in ensuring robust security. In [48], the researchers introduced a scan-based permutation and substitution technique for video encryption. According to [49], a probabilistic approach, leveraging a logistic adjusted sine map [12], was suggested for keyframe encryption. This method is designed to be energy-efficient, making it suitable for Internet of Things (IoT) applications. In [11], the authors introduced an innovative chaotic map aimed at producing intricate chaotic dynamics. Their approach involved coupling Sine and Logistic maps, followed by the application of a sine function to the combined output. In our research, we utilize the approach from [11] to generate the chaotic sequence. Valli et al. proposed two methods in their study [40]. The first method is founded on a 12-dimensional chaotic map, while the second employs the Ikeda delay differential equation. The authors assert that their video encryption technique, which operates on a frame-by-frame basis, is compatible with real-time applications.. Socek et al. in [38] proposed correlation preserving video encryption by using permutation. A video encryption method that is independent of compression (CVES) was introduced by Li et al. [50]. This method employs various chaotic systems to encrypt videos on a frame-by-frame basis. In a separate work [41], the authors introduced a puzzle-based algorithm for encrypting video that is also compression-independent and is designed to support real-time

applications. Authors have utilized a high-dimensional Lorenz chaotic system to encrypt video data on a frame-by-frame basis, as noted in [42]. However, merely altering positions does not ensure enhanced security. To address this, a method [51] is introduced using public key encryption (PKE) for images and videos leveraging Chebyshev maps. In [52], the authors introduced a three-tier security approach to video encryption employing a blend of Logistic and Tent chaotic maps. Their method incorporates both confusion and diffusion at the block level. Two methods, which are independent of compression, have been introduced in [53] for video encryption utilizing four chaotic maps. Any change in an original video frame can only affect the corresponding encrypted frame, as most of the above-mentioned methods use frame-by-frame encryption. Any advanced video encryption technique that relies on a higher-dimensional chaotic map demands numerous parameters to generate the chaotic sequence. However, the requirement for many parameters translates to larger key sizes, complicating key management. The objective is to introduce a reliable and efficient encryption technique tailored for uncompressed video transmission. This method incorporates an inter-frame dependent shuffling strategy, ensuring that alterations in a single original frame influence all ensuing encrypted frames, enhancing overall robustness. Our approach is secure and advantageous due to the need for exchanging fewer secret parameters securely between the sender and receiver prior to the confidential transmission of the video.

2.1.3 Proposed Framework

This section elaborates comprehensively on the method we propose, divided into three segments. Initially, we describe how to generate a pseudorandom chaotic sequence. Next, we introduce our method of confusion based on sort index, illustrating the shuffling of pixel positions across various frames. Lastly, we delve into the block-level diffusion technique applied to the confused frames. Fig. 2.1 displays the proposed framework, and the encryption process is executed through two rounds of confusion and subsequently two rounds of block-level diffusion.

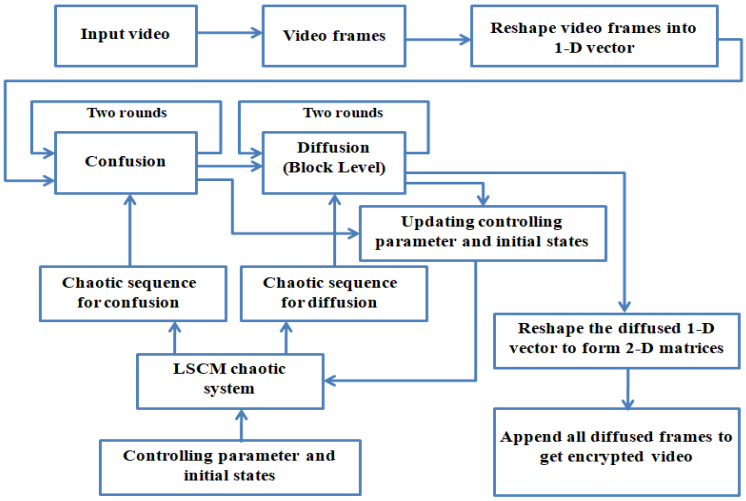


Figure 2.1: Flow chart of the proposed video encryption technique.

2.1.3.1 Pseudorandom Sequence Generation

A 2-D logistic-sine-coupling map (LSCM) [11] is utilized for sequence generation. This map is chosen because of its high sensitivity to initial conditions and control parameters as well as its intricate chaotic dynamics, which are highly efficient. Its extensive chaotic and hyperchaotic ranges yield more random sequences. To establish a connection between the generated sequence and the original frame, the total sum of all the pixel values of the original frame has been calculated. Utilizing the Diehard statistical suite [54], we conducted a randomness evaluation on a generated chaotic sequence, with the results detailed in Table 2.1. For the randomness assessment, we utilized the parameters $x = 0.5684, y = 0.4278, \theta = 0.9121$. In this experiment, Algorithm-1 produced a sequence comprising 16777216 bytes, which was then subjected to analysis using the Diehard statistical tests. The LSCM is mathematically represented by the equation:

$$\begin{aligned} x_{i+1} &= \sin(\pi(4\theta x_i(1-x_i) + (1-\theta)\sin(\pi y_i))) \\ y_{i+1} &= \sin(\pi(4\theta y_i(1-y_i) + (1-\theta)\sin(\pi x_i))) \end{aligned} \tag{2.1}$$

where θ is the controlling parameter and $(x, y, \theta) \in [0, 1]$.

Algorithm 1 Generation of pseudorandom chaotic sequence

Input: (F, x_0, y_0, θ), where F is input video frame of size [1,3MN]
Output: (Chaoticseq)

Size of each frame $\leftarrow [M, N, 3]$
Length of Chaotic sequence $\leftarrow [1, 3MN]$
$S = \sum_{i=1}^{3MN} F(i)$
if $S = 0$
$t \leftarrow 0$
else
$t = \frac{\sqrt{S}+5}{(S+255)}$
end
$x_1 = x_0 + t, y_1 = y_0 + t, \theta = \theta + t;$
for i=1 to 3MN **do**
$x_{i+1} = \sin(\pi(4\theta x_i(1 - x_i) + (1 - \theta)\sin(\pi y_i)))$
$y_{i+1} = \sin(\pi(4\theta y_i(1 - y_i) + (1 - \theta)\sin(\pi x_i)))$
$Chaoticseq(i) = floor(10^{10} \times (x_{i+1} + y_{i+1}))$ mod 256
end for

Table 2.1: Randomness Test using Diehard Suite

Statistical Test	**P value**	**Result**
Birthday spacings	0.135756	Pass
Overlapping 5-permutation	0.589286	Pass
Binary rank for 31x31 matrices	0.613440	Pass
Binary rank for 32x32 matrices	0.378504	Pass
Binary rank for 6x8 matrices	0.080413	Pass
Bitstream	0.46525	Pass
Overlapping-Pairs-Sparse-Occupancy	0.582900	Pass
Overlapping-Quadruples-Sparse-Occupancy	0.518500	Pass
DNA	0.456200	Pass
Count-the-1's on a stream of bytes	0.435546	Pass
Count-the-1's for specific bytes	0.123514	Pass
Parking lot	0 .581103	Pass
Minimum distance	0.350740	Pass
3D spheres	0.278296	Pass
Sqeeze	0.214191	Pass
Overlapping sums	0.798832	Pass
Runs	0.907553	Pass
Craps	0.536751	Pass

2.1.3.2 Sort Index Based Shuffling

In the technique we suggest, the secret key for video encryption consists of the initial conditions and control parameters $(x_0, y_0, \theta_0, m_1, n_1, \theta_1)$. Initially, each color video frame of dimensions [M, N, 3] is reorganized into a one-dimensional array of size [1, 3MN]. Subsequently, Algorithm-1 is employed to generate a chaotic sequence, denoted as D1, in the following manner:

$$D1 = Chaoticseq(\mathbf{0}, x_0, y_0, \theta_0) \tag{2.2}$$

Let **0** represent a zero vector with a length equivalent to that of a single video frame. The initial frame is rearranged based on the sort index derived from the chaotic sequence D1. Using Algorithm-1, another chaotic sequence is produced and referred to as D2 in the following manner:

$$D2 = Chaoticseq(\text{Previous confused frame}, x_0, y_0, \theta_0) \tag{2.3}$$

Then D2 is modified to get another sequence denoted by D3 as follows:

$$D3 = ((\text{Previous confused frame+D2}) \mod 256) \tag{2.4}$$

Finally, the remaining frames are shuffled by the sort index of D3 using Eq.2.3 and Eq.2.4. Second round of confusion for all frames are completed by Eq.2.2, Eq.2.3 and Eq.2.4 taking secret parameter as (m_1, n_1, θ_1).

2.1.3.3 Block Level Diffusion

After the processes of confusion and diffusion, let CF and C represent the shuffled and scrambled frame arrays, respectively. The scrambling of frames is achieved through block level diffusion. Each shuffled frame array is partitioned into P blocks, each with a block length denoted by L. The length L is selected such that the total number of pixels in a frame, 3MN, divides evenly by L. For each block, a new chaotic sequence, adjusting the sequence length from 3MN to L, is produced using Algorithm-1. The block-level diffusion process for each video frame array in first

round is illustrated as:

$$C_{i(Block)} = \begin{cases} CF_{i(Block)} \oplus CF_{(LastBlock)} \oplus S_1, & for 1^{st} Block \\ C_{i-1(Block)} \oplus CF_{i(Block)} \oplus S_2, & \text{for other Blocks} \end{cases} \tag{2.5}$$

where S_1 and S_2 are generated by Eq.2.6 and Eq.2.7 respectively using Algorithm-1 and are given by:

$$S_1 = Chaoticseq(\mathbf{0}, x_0, y_0, \theta_0) \tag{2.6}$$

$$S_2 = Chaoticseq(\text{Previous diffused block}, x_0, y_0, \theta_0) \tag{2.7}$$

The second round of diffusion for all frame arrays is completed by Eq.2.5 to Eq.2.7 considering secret parameter as (m_1, n_1, θ_1).Ultimately, every diffused array is converted into 2D RGB matrices, and all these diffused frames are combined to create an encrypted video. The process of decrypting this encrypted video involves performing two rounds of inverse block diffusion and then two rounds of inverse confusion operations.

2.1.3.4 Time Complexity

A video frame contains r pixels for every color element, where $r = MN$ and M and N represent the frame's number of rows and columns, respectively. Suppose the video is made up of α frames. The suggested encryption method involves three stages. The first stage generates distinct chaotic sequences for various frames utilizing the LSCM map. So time complexity of this step is $O(r)$. Secondly, each frame is shuffled by a sort index-based permutation. So this step is $O(r^2)$. Thirdly, each block in shuffled frame is scrambled by bitwise xor with chaotic sequence and previous scrambled block. Let P be number of blocks found when dividing r by length of each block(L). A unique chaotic sequence is generated for each block using previous scrambled block with $O(L)$ time. Time complexity for scrambling, P number of blocks is of $O(PL)$. So, the overall complexity of each encrypted frame is $(O(r) + O(r^2) + O(PL)) \approx$

Algorithm 2 Encryption of video at frame level

Input: $(F, P, x_0, y_0, \theta_0, m_1, n_1, \theta_1)$, where F is input video frame of size [1,3MN] and P is the number of blocks in a frame whose length is L.

Output: (Encrypted video)

for i=1 to Last frame **do**
  if $i = 1$
  $D1 = Chaoticseq(0, x_0, y_0, \theta_0)$
  $D1^{'} = sort(D1)$
  $CF_{(i,:)} = F_{(i,:)}(D1')$
  else
  $D2 = Chaoticseq(CF_{(i-1,:)}, x_0, y_0, \theta_0)$
  $D = CF_{(i-1}, x_0, y_0, \theta_0)$
  $D3 = (CF_{i-1} + D2) \mod 256$
  $D3 = (CF_{(i-1,:)} + D2) \mod 256$
  $D3^{'} = sort(D3)$
  $CF_{(i,:)} = F_{(i,:)}(D3')$
  end if
end for
For second round of shuffling is completed by repeating step.1 to step.12 taking secret parameter as (m_1, n_1, θ_1)
for i=1 to Last Frame **do**
  for j=1 to Last block **do**
    if $j = 1$
    $S1 = Chaoticseq(0, x_0, y_0, \theta_0)$
    $C_{(i,(j-1)*L+1:j*L)} = CF_{(i,(P-1)*L+1:P*L)} \oplus S1 \oplus CF_{(i,(j-1)*L+1:j*L)}$
    else
    $S2 = Chaoticseq(C_{(i,(j-2)*L+1:(j-1)*L}, x_0, y_0, \theta_0)$
    $C_{(i,(j-1)*L+1:j*L)} = C_{(i,(j-2)*L+1:(j-1)*L)} \oplus S2 \oplus CF_{(i,(j-1)*L+1:j*L)}$
    end if
  end for
end for

$O(r^2)$(as $P, L << r^2$). For α number of frames in a video, the overall complexity of our algorithm is $O(\alpha r^2)$.

2.1.4 Experimental Results

We tested our suggested technique on a collection of standard videos referenced in [40]. This set consists of Rhinos.avi ($240 \times 320 \times 3$), Train.avi ($192 \times 352 \times 3$), Viptrain.avi ($240 \times 360 \times 3$), and Flamingo.avi ($192 \times 352 \times 3$).The experiments are conducted on a desktop computer equipped with an Intel(R) Core (TM) i5-9330H processor running at a clock speed of 2.40 GHz and with 8 GB RAM. For

the implementation process, MATLAB 2020a is utilized. We have experimentally set $x_0 = 0.5684, y_0 = 0.4278, \theta_0 = 0.9121, m_1 = 0.3255, n_1 = 0.5258, \theta_1 = 0.9091$ as secret key and block length as 256 for block level diffusion. In this section, We compare our experimental results with three existing compression independent video encryption methods [40] and [52].

2.1.4.1 Correlation Analysis

Correlation coefficient between two pixels p and q, denoted by $CC_{p,q}$ is given by:

$$CC_{pq} = \frac{cov(p,q)}{\sqrt{D(p) \times D(q)}} \tag{2.8}$$

Where $cov(p,q)$ is the covariance between p and q, $E(p), E(q)$ and $D(p), D(q)$ are the expectation and variance of variable p and q respectively. The value of CC_{pq} is close to 1 for original video frames and is near to 0 for encrypted frames. In Table 2.2 ,we have compared average correlation coefficient of the videos before and after encryption of two adjacent pixels along horizontal, vertical and diagonal directions with other existing methods. In our method the encrypted frames have correlation coefficients nearly equal to zero. We have randomly selected 3000 pixels in input video frames and their respective adjacent pixels in the horizontal, vertical, and diagonal directions for finding correlation coefficient.

2.1.4.2 Entropy Analysis

Entropy serves as a metric for measuring randomness. Greater entropy signifies increased uncertainty. The entropy of encrypted frames tends to approach 8, suggesting these frames exhibit a high degree of randomness. Moreover, we evaluated our approach on frames entirely composed of pixels at values 0 and 255, which resulted in encryption that appears entirely random. The entropy for a video frame can be calculated as:

$$H(k) = -\sum_{i=1}^{255} P(k_i) \log_2 P(k_i) \tag{2.9}$$

Table 2.2: Mean Correlation Coefficient

Methods	Video Files	CC before encryption			CC after encryption		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
12D Map [40]	**Rhinos**	0.9866	0.9765	0.9673	0.0196	0.0192	0.0138
	Train	0.937	0.9088	0.8735	0.0159	0.0195	0.0191
	Flamingo	0.9753	0.9697	0.948	0.0155	0.0146	0.0171
	Viptrain	0.9517	0.9497	0.9029	0.0203	0.0175	0.0224
Ikeda DDE [40]	**Rhinos**	0.9866	0.9765	0.9673	0.0181	0.0140	0.0107
	Train	0.937	0.9088	0.8735	0.0154	0.0105	0.0121
	Flamingo	0.9753	0.9697	0.948	0.0150	0.014	0.0148
	Viptrain	0.9517	0.9497	0.9029	0.0176	0.0147	0.0158
[55]	**Rhinos**	0.9866	0.9765	0.9673	0.0181	0.0140	0.0107
	Train	0.937	0.9088	0.8735	0.0154	0.0105	0.0121
	Flamingo	0.9753	0.9697	0.948	0.0150	0.014	0.0148
	Viptrain	0.9517	0.9497	0.9029	0.0176	0.0147	0.0158
8D Map [40]	**Rhinos**	0.9866	0.9765	0.9673	0.0254	0.0184	0.0135
	Train	0.937	0.9088	0.8735	0.0186	0.0179	0.0155
	Flamingo	0.9753	0.9697	0.948	0.0227	0.015	0.0159
	Viptrain	0.9517	0.9497	0.9029	0.0192	0.0133	0.0165
Proposed Method	**Rhinos**	0.9839	0.9741	0.9735	0.0040	0.0080	0.0127
	Train	0.9400	0.9287	0.8822	0.0063	0.0154	0.0126
	Flamingo	0.9783	0.9767	0.9568	0.0074	0.0061	0.0094
	Viptrain	0.9505	0.9566	0.9197	0.0020	0.0048	0.0114

Table 2.3: Entropy Value

Method	Video files	Entropy of original frame	Entropy of encrypted frame
Proposed method	**Rhinos**	7.0769	7.9992
	Train	7.2074	7.9998
	Flamingo	6.3881	7.9994
	Viptrain	7.4370	7.9993
	Frame having all value 0	0	7.9993
	Frame having all value 255	0	7.9992

where H(k) is the entropy,$P(k_i)$ indicates the probability of symbol k_i. We have shown the entropy values of original video frame(4^{th}) and encrypted video frame(4^{th}) in Table 2.3.

2.1.4.3 Histogram Analysis

Fig. 2.2 presents a histogram comparison of the original video frame and its encrypted version, focusing on the R, G, and B color components of the Rhinos video's 4th frame. The disparities between the two histograms are considerable, as the encrypted frame shows an almost even distribution. This evenness suggests that the

proposed encryption method effectively withstands statistical attacks.

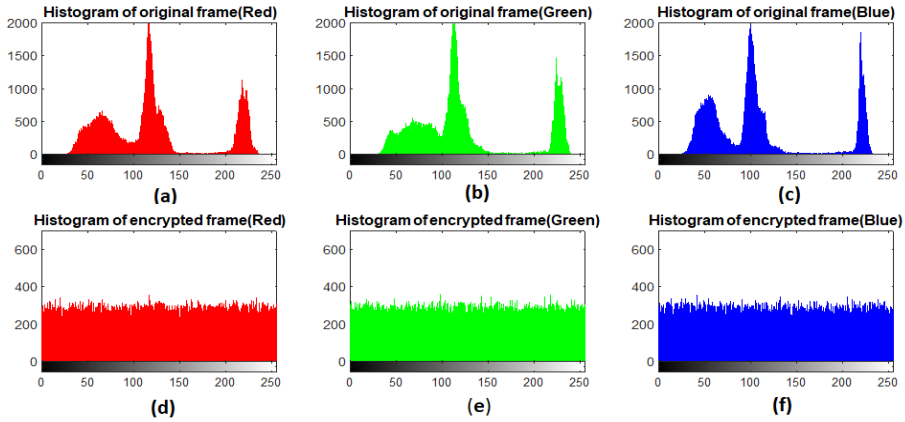


Figure 2.2: Histograms of color components showing differences between (a-c) original and (d-f) encrypted frame

2.1.4.4 NPCR and UACI Analysis

The number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) values have to be greater than ideal values of 99.5693 and 33.4635 respectively to defend differential attack. The expressions for NPCR and UACI as follows:

$$NPCR(C_1, C_2) = \sum_{i,j} \frac{W(i,j)}{M \times N} \times 100\% \tag{2.10}$$

$$UACI(C_1, C_2) = \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255 \times M \times N} \times 100\% \tag{2.11}$$

where C_1, C_2 are two encrypted frames generated from same video frame and secret key but changing a single pixel value. $M \times N$ is the size of video frames and $W(i,j) = 1$ if $C_1 \neq C_2$ and is 0 otherwise. We have used [56] for testing purpose to show that our scheme can resist differential attack. Table 2.4 presents the NPCR and UACI outcomes. We conducted experiments under various conditions, including altering a single bit in the pixel value of an original video frame, and modifying the initial conditions and control parameters. In all scenarios, our approach achieves

NPCR and UACI values surpassing the threshold, providing robust security against differential attacks.

2.1.4.5 Mean Encryption Time

In Table 2.5, we present a comparison of the average encryption time for our technique against some established methods [40], [38]. Our approach is faster than the method in [40]. As the block length grows, our method achieves quicker encryption times. However, there exists a trade-off between block length and the average encryption time.

2.1.4.6 Analysis of Secret Key

It's common to use a key space of 2^{128} to guard against exhaustive attacks in encryption techniques. In the security algorithm we've developed, we utilize six key parameters: $x_0, y_0, \theta_0, m_1, n_1, \theta_1$. Each of these parameters is defined with a precision of 10^{-15}. Consequently, the entire secret key space amounts to $10^{15*6} = 10^{90}$. This key space is roughly equivalent to $10^{90} \approx 256 \times 2^{256}$, providing ample protection against brute force attacks. In Table 2.6, we present a comparison of the key space of our approach with several recent methodologies. An insignificant alteration in the

Table 2.4: Comparison of NPCR and UACI values of encrypted video frames

Methods	**Rhinos video**		**Train video**		**Flamingo video**		**Viptrain video**	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
12D Map [40]	99.44	33.485	99.36	33.499	99.37	33.4985	99.34	33.4596
Ikeda DDE [40]	99.95	33.5914	99.94	33.5065	99.94	33.5886	99.95	33.599
[55]	99.51	33.54	99.61	33.50	99.52	33.52	99.58	33.62
8D Map [40]	99.32	33.44	99.36	33.46	99.36	33.4077	99.28	33.4427
Proposed Method	99.64	33.5021	99.62	33.4918	99.62	33.5088	99.63	33.4611

Table 2.5: Comparison of mean encryption time in seconds

Methods	**Video Files**			
	Rhinos	Train	Flamingo	Viptrain
12D Map [40]	1.7964	1.6493	0.8402	2.3137
Ikeda [40]	1.2147	0.9908	1.1124	1.3574
[55]	0.5403	0.5178	0.5982	0.4723
8D Map [40]	0.9124	0.8631	0.8062	1.0588
Proposed Method	0.7946	0.8548	0.8538	0.9890

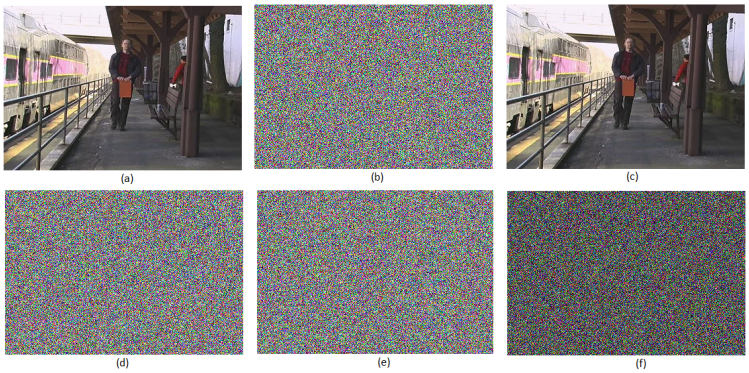


Figure 2.3: (a) Original frame No. 4 from the Viptrain video; (b) corresponding encrypted frame using keys (0.5684, 0.4278, 0.9121, 0.3255, 0.5258, 0.9091); (c) decrypted frame with the correct keys; (d) decrypted frame with a slight key variation ($0.5684 + 10^{-15}$, 0.4278, 0.9121, 0.3255, 0.5258, 0.9091); (e) decrypted frame with another key variation (0.5684, $0.4278 + 10^{-15}$, 0.9121, 0.3255, 0.5258, 0.9091); (f) difference frame between (d) and (e).

key renders the decryption process incapable of retrieving the original video frames, thereby ensuring the proposed method's resilience to differential attacks. Figures 2.3 and 2.4 demonstrate how even a slight deviation in the secret key can render decryption unsuccessful with our method. Decryption is only achievable when the secret keys of both the sender and receiver align perfectly..

Table 2.6: Comparison of key space

Methods	Keyspace
12D Map [40]	3072×2^{128}
Ikeda DDE [40]	49152×2^{128}
[42]	2^{128}
[55]	2^{212}
8D Map [40]	8×2^{128}
Proposed Method	256×2^{256}

2.1.4.7 Robustness Against Noise

Encrypted video frames undergo with several types of noise: salt and pepper noise with a density of 0.05, Gaussian noise characterized by a mean of 0 and a variance of 0.03, and speckle noise also with a density of 0.05. Due to these noise additions, the decrypted video—assuming exact key utilization—retains a noisy condition, alerting the receiver to potential data tampering during transmission. For heightened

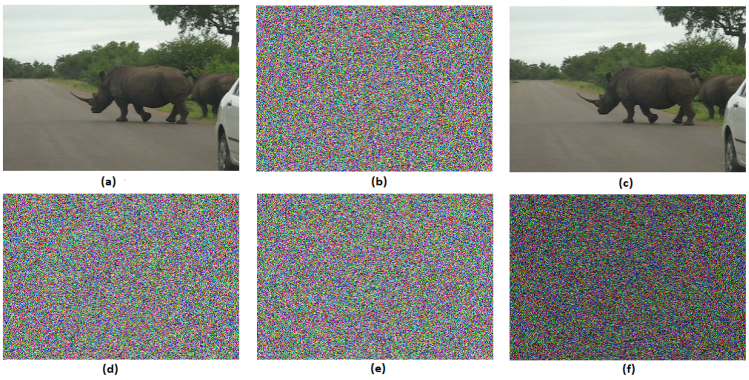


Figure 2.4: (a) Original frame No. 4 from the Rhinos video; (b) corresponding encrypted frame using keys (0.5684, 0.4278, 0.9121, 0.3255, 0.5258, 0.9091); (c) decrypted frame with the correct keys; (d) decrypted frame with a slight key variation ($0.5684 + 10^{-15}$, 0.4278, 0.9121, 0.3255, 0.5258, 0.9091); (e) decrypted frame with another key variation (0.5684, $0.4278 + 10^{-15}$, 0.9121, 0.3255, 0.5258, 0.9091); (f) difference frame between (d) and (e).

security, it is essential that the peak signal-to-noise ratio (PSNR) of the encrypted video remains low. Figure 2.5 illustrates the average PSNR values for the first fifteen encrypted frames. The PSNR value for the decrypted video should be high so that it can resist different channel noise. We have shown decrypted video frames after the addition of different noises in encrypted video frames in Fig. 2.6. The PSNR is computed as:

$$PSNR_{dB} = 10 \times \log_{10} \frac{255^2}{MSE} \tag{2.12}$$

where MSE is the mean squared error given as

$$MSE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \frac{|I(i,j) - C(i,j)|^2}{M \times N} \tag{2.13}$$

where I is original frame and C is encrypted frame.

2.1.4.8 Security Analysis

For an encryption technique to be deemed computationally secure, it must fulfill two criteria: The expense of decrypting the cipher should exceed the actual worth of the protected data, and the time required to breach the encryption must surpass the data's operational life span. Our suggested approach meets these criteria. Security

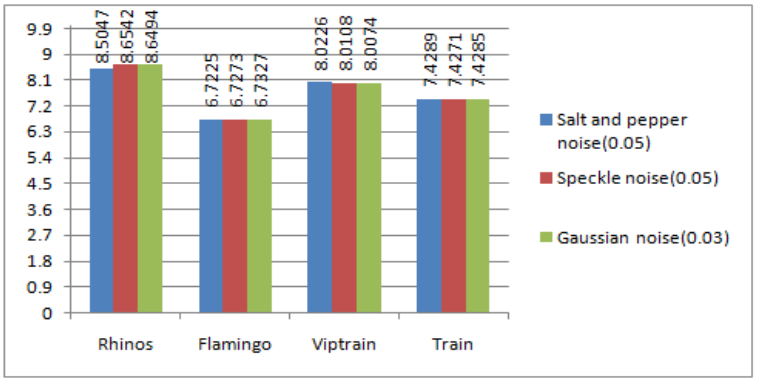


Figure 2.5: Mean PSNR of encrypted video

evaluation has been conducted by evaluating known and chosen-plaintext attacks.

When facing a known-plaintext attack, the attacker is privy to the plaintext, ciphertext, and the encryption algorithm. The adversary must generate distinct security keys for different frames. The security key for each frame is derived by summing all the pixels from the previously shuffled frame. Our proposed method offers a keyspace of 10^{90}, entailing that breaking the encryption requires 10^{90} possible secure key combinations. Additionally, the security key varies with each iteration, significantly complicating the process of key extraction. Therefore, our proposed approach is robust against known-plaintext attacks.

In a chosen-plaintext attack scenario, it is presumed that the adversary has access to one pair of a plain frame and its corresponding encrypted frame, denoted as (I1, C1). The attacker is also able to select a second frame, I2, which has the same dimensions as the original but differs only in the first row by one unit. Let T1 and T2 represent the chaotic sequences for shuffling I1 and I2 respectively. Since the sum of the pixel values in I1 and I2 are different, T1 and T2 are also distinct, leading to

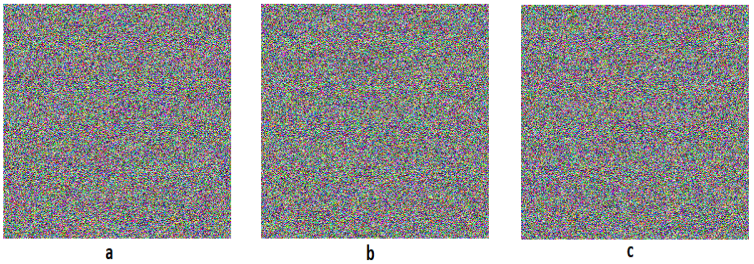


Figure 2.6: Decrypted video frame under:(a) salt and pepper noise (b) speckle noise (c) Gaussian noise

different sort indices. Consequently, I1 and I2 are shuffled differently. During the scrambling process, a unique sequence is produced for each block, based on the sum of the preceding scrambled block, resulting in distinct scrambled frames for I1 and I2. The chaotic sequence for shuffling each frame considers the previously shuffled frame, while the scrambling process includes the prior scrambled block, enhancing security and robustness against a chosen-plaintext attack.

2.1.4.9 External Comparisons

Our suggested method involves encrypting the entire video as opposed to just selective or partial encryption. Table 2.7 offers a summary and comparison of the performance of various video encryption techniques. We evaluated our experimental findings against three methods as described in [40] and [38]. Our findings exhibit superior performance regarding the average correlation coefficient, key space, mean encryption time, entropy, and PSNR of the encrypted video. Thus, the proposed approach is effective and suitable for real-time video security applications. The NPCR and UACI values of our method are higher as compared to 12D map [40], [38]. In their first method (12D chaotic map) [40], twelve initial conditions were used. So it is difficult to communicate all the initial values over the channel if sufficient bandwidth is not available. We have adopted six initial states in our approach, which is more bandwidth-efficient than the 12D chaotic map technique. The histogram of the encrypted video frames achieves greater uniformity. Altering a pixel value by 1 bit in any position within any frame using our proposed method impacts numerous pixels across all frames, thereby enhancing NPCR and UACI. Elevated security becomes crucial for videos stored over extended durations due to the ample time available to attackers. Our method is exceptionally sensitive to even minor alterations in the original video frames, initial states, and controlling parameters, ensuring robustness, efficiency, and superior security. The method also yields higher entropy, signifying that encrypted video frames remain highly unpredictable.

Table 2.7: Performance comparison of different methods

Reference	Process used	Type of encryption	Security	Application
12D [40]	Confusion, Diffusion	Complete	Moderate	Wireless communication
Ikeda DDE [40]	Confusion, Diffusion	Complete	High	Video storage and communication
[42]	Confusion, Diffusion	Complete	Moderate	Video communication
[51]	Random Modulation	Complete	Low	Wireless communication
[47]	Confusion, Diffusion	Complete	High	Video storage
Proposed method	Confusion, Diffusion	Complete	High	Video stoarge and communication

2.1.5 Discussion

This work introduces a video encryption technique that operates independently of compression at the frame level. For each frame, a fresh chaotic sequence is generated using the preceding frame and the previously created chaotic sequence. A secure sorting permutation approach is utilized to shuffle the video frames. Furthermore, a distinct chaotic sequence is generated for each block within the frame, and a block-level diffusion technique is applied to scramble the video frames. Experimental comparisons using various measures clearly establish the superiority of the proposed method over some of the existing approaches. Additionally, we have evaluated the robustness of our proposed approach against brute-force attacks, key sensitivity for decryption, PSNR of encrypted videos, entropy analysis, differential attacks, and assessments for known and chosen-plaintext attacks. The thorough evaluation unmistakably demonstrates the high security of our video encryption technique. In the future, we plan to investigate encryption techniques that depend on video compression.

2.2 A Fast and Secure Video Encryption using Divide-and-Conquer approach

2.2.1 Introduction

Video encryption using chaotic map has evolved as an important research topic in secure video data storage and transmission. In [32], authors discussed about video encryption with and without compression. Chaotic maps are popular for generating pseudorandom sequences as they are highly unpredictable, sensitive to initial states and control parameters and are ergodic in nature. Many chaotic maps like Sine, Logistic, Tent that have been used for encrypting the images and video suffer from lower chaotic ranges. So, a better chaotic map is necessary for ensuring more secure video encryption. Further, existing chaos based uncompressed video encryption schemes do not necessarily exploit the temporal redundancy in a video [40, 53]. Rather, they simply treat each frame as an image for encryption. As a video deals with huge amount of data, it needs to be processed efficiently while preventing illegitimate access and preserving infringement confidentiality. Clearly, direct encryption of all the frames of a video taken together could be computationally very expensive. So, some efficient algorithmic approach and use of temporal redundancy are required.

In this work, we propose a fast and secure compression independent video encryption scheme. Our solution is computationally efficient as we use a divide-and-conquer type approach and exploit temporal redundancy. Better security is achieved through a new chaotic map termed as Logistic, Tent and Infinite Collapse Map (LT-ICM) with a higher chaotic range and more complex behavior. We employ a divide-and-conquer type approach by dividing a video into shots and then process the frames within each shot separately. Shots are obtained by finding the changes between two consecutive frames using the sum squared difference (SSD) measure [57]. Note that the frames within each shot are highly correlated. So, we exploit temporal redundancy by obtaining difference frames within a shot. Next, we threshold non-zero pixels in the difference frames to further reduce the computational load for processing. The resulting set of sparse pixels are represented using a 4-tuple with row

position, column position, frame number and intensity values. We use sorting based random permutation and substitution in the first level of encryption. For the second level of encryption, we use only XOR operation which makes encryption process fast. These steps are repeated for each shot. The main contributions of our video encryption scheme are now highlighted below:

1. We generate a new pseudorandom chaotic sequence using our LT-ICM chaotic map. The proposed map has better chaotic range, complex behaviour, unpredictability, ergodicity, and sensitivity towards initial values and controlling parameters.

2. We apply a divide-and-conquer type of approach where instead of processing all the frames of a video together, we process frames in a shot-wise manner. Further, a sparse representation of pixels within the frames in a shot are adopted by exploiting temporal redundancy. These steps make the proposed encryption scheme fast.

The remainder of the work is structured as follows: In section 2.2.2, we discuss the related work. In section 2.2.3, we describe our proposed video encryption architecture, chaotic sequence generation, and its performance analysis. Section 2.2.5 presents the experimental results with analysis. The work is discussed in Section 2.2.6.

2.2.2 Related Work

Video encryption without compression has many advantages over that of encryption with compression. While encrypting video with codec video data need to satisfy different requirement like format complaint, codec complaint, quality of video to be perceivable, minimum time for encryption and fixed bitrates. Authors in [53], used four chaotic maps for encrypting video in frame level. In [40] authors have presented their video encryption scheme by using substitution box operating on cipher block chaning mode where they have used 8, 12 dimensional chaotic map and Ikeda delay differential system for chaotic sequence generation. Scan based

permutation is addressed in [48] for encrypting images and videos. Sethi et al. [58] presented a frame level encryption for uncompressed video using Logistic Sine Coupling Map. But they have not considered temporal redundancy and their method takes more time than that of our proposed method. Authors in [52] implemented a video encryption scheme employing combined Logistic and Tent map. In [42], authors presented video encryption using Logistic and Lorenz chaotic map.

In [59], authors generated quantum walks based substitution box for video encryption. Song et al. [60] proposed a quantum video encryption scheme employing improved Logistic map. A lightweight probabilistic keyframe encryption is addressed in [49] which woks in IoT system. Compressive sensing based efficient cryptosystem has been addressed in [34] which supports IoT environment. Chaotic system based on infinite collapse map are presented in [61]. There are some periodic windows and discontinuity in chaotic range. We generate a new chaotic map LT-ICM which overcomes these limitations. Most of the video encryption techniques addressed above are not that fast and also they have not considered the temporal redundancy. Our proposed divide-and-conquer compression independent video encryption method is fast as we deal with less number of non-zero pixel values in difference frames within shots. Also, our encryption scheme is secure as LT-ICM map is highly unpredictable in the chaotic range.

2.2.3 Proposed Architectures

In this section, we describe the details of our proposed method. This section consists of three parts. The first part consists of generation of the chaotic sequence using LT-ICM map and its performance analysis. The second part, deals with video to shot conversion, de-correlation of frames present within each shot and conversion of shot into sparse form. The third part deals with conversion of chaotic sequence based permuted and substituted sparse data into frame level data for each shot. After that update frames present in each shot are XOR-ed with a chaotic sequence which depends on initial state, controlling parameter and previous encrypted frame. The complete block diagram of video encryption scheme is presented in Fig. 2.7.

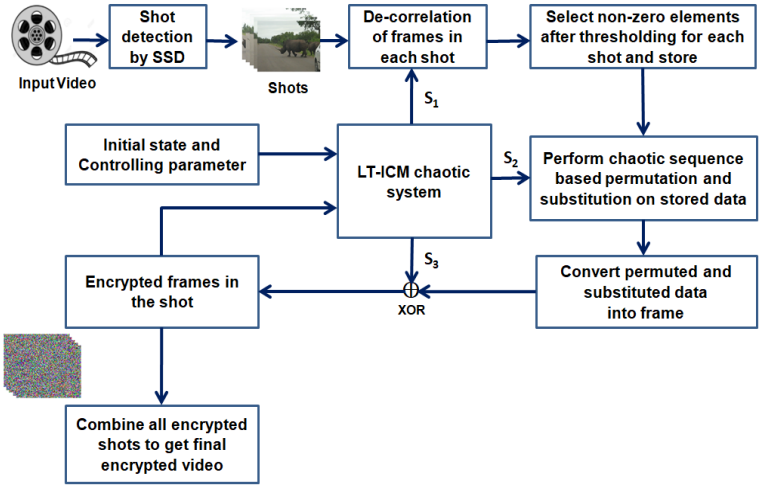


Figure 2.7: Proposed block diagram of video encryption scheme

2.2.3.1 Pseudo-random Sequence Generation(PRSG)

Logistic map, Tent map and Iterative chaotic map with infinite collapse are common 1D chaotic maps. Mathematically, Logistic, Tent and ICM maps are defined in equation 2.14, 2.15 and 2.16 respectively.

$$x_{i+1} = r_1 x_i (1 - x_i) \tag{2.14}$$

$$x_{i+1} = \begin{cases} 2r_2 x_i & \text{if } x_i < 0.5 \\ 2r_2(1 - x_i) & \text{if } x_i \geq 0.5 \end{cases} \tag{2.15}$$

$$x_{i+1} = \sin(r_3 / x_i) \tag{2.16}$$

where r_1,r_2,r_3 are the controlling parameters for Logistic, Tent and ICM map respectively and $r_1 \in [0, 4]$, $r_2 \in [0, 1]$, $r_3 > 0$.

The chaotic range of former two chaotic maps is narrow. The Logistic and Tent map have chaotic behaviour when $r_1 \in [3.57, 4]$ and $r_2 \in [0.5, 1]$. A chaotic map is proposed using Logistic, Tent and ICM map(LT-ICM). Mathematically, the LT-ICM

can be represented as:

$$x_{i+1} = \sin(r_3/(P(r_1, x_i) + Q(r_2, x_i) + 0.5) \tag{2.17}$$

As indicated in Eq.2.17, the LT-ICM first adds the two base maps $P(r_1, x_i)$ and $Q(r_2, x_i)$ then shift the result by 0.5 and finally apply to the ICM map to generate the LT-ICM chaotic map output. The combination operation can effectively shuffle the chaos dynamics of the two base maps, and the ICM map produce very complex nonlinearity. Thus, the new chaotic map produced by the LT-ICM has complex behavior. We have considered $P(r_1, x(i))$ as Logistic map, $Q(r_2, x(i))$ as Tent map. For simplification, we consider $r_2 = 1 - r_1$ and $r_3 = r_1$. The generated LT-ICM chaotic map has better chaotic range, unpredictability and ergodicity than base maps. After modification, the LT-ICM can be represented as:

$$x_{i+1} = \begin{cases} \sin(r_1/(2x_i r_1 + (1 - r_1)x_i(1 - x_i) + 0.5)) & \text{if } x_i < 0.5 \\ \sin(r_1/(2(1 - x_i)r_1 + (1 - r_1)x_i(1 - x_i) + 0.5)) & \text{if } x_i \geq 0.5 \end{cases} \tag{2.18}$$

2.2.3.2 Performance Analysis of Chaotic Sequence

For evaluating LT-ICM, we select different measures like bifurcation diagram, Lyapunov exponent. We have also tested the sensitivity to initial condition and controlling parameter for the proposed map by applying a minor change in both parameters. The correlation coefficient value for two sequences using LT-ICM map with minor change in controlling parameter and initial condition is nearer to zero. So the two sequences are different with any minor change in parameter and initial state. The entropy of proposed map is high for the chaotic range where lyapunov exponent is non-zero, which indicates the generated sequence is unpredictable.

2.2.3.3 Bifurcation Diagram

For good chaotic performance, the attractor has to occupy the maximum regions in phase space. We consider 0.3 as the initial state and iterate 20000 times for every map. Next, we plot all points in Phase space. We compare the result with base

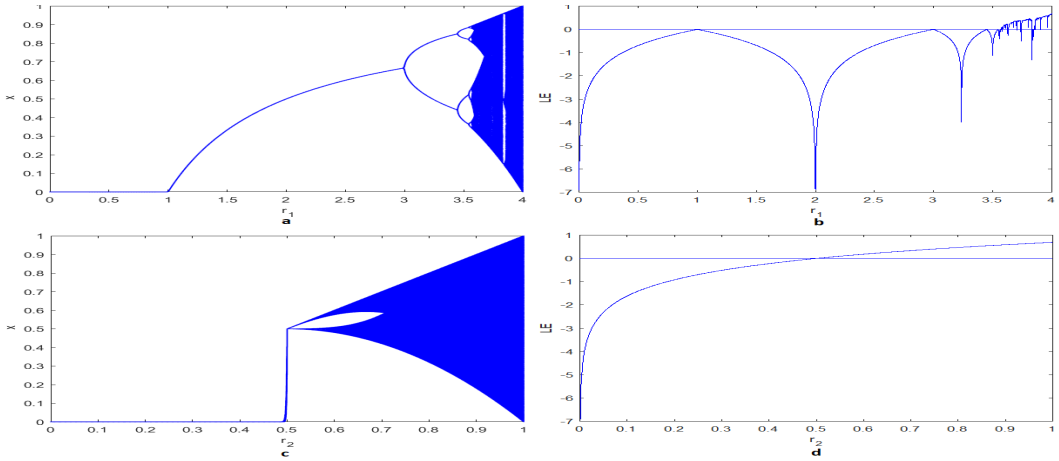


Figure 2.8: Bifurcation diagram and LE of (a, b) Logistic map, (c, d) Tent map

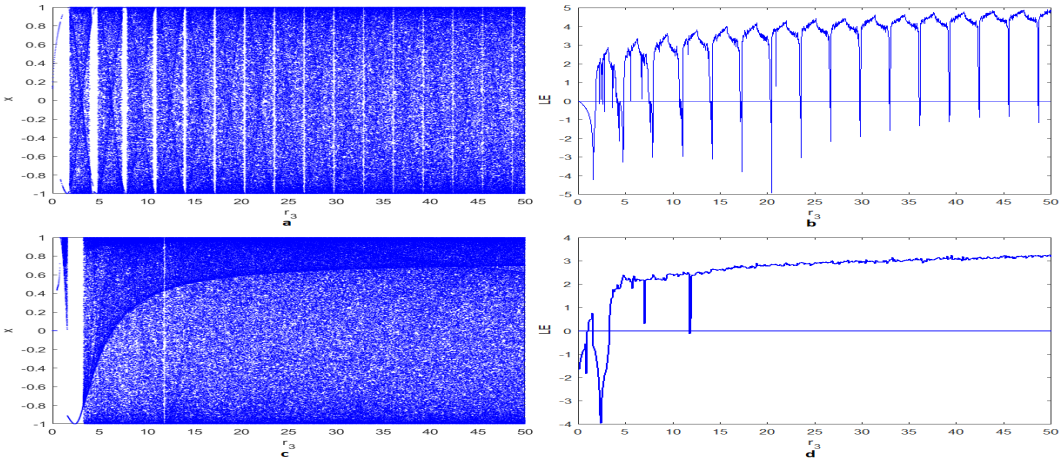


Figure 2.9: Bifurcation diagram and LE of (a, b) ICM map, (c, d) LT-ICM map

maps. Fig. 2.8 shows the bifurcation diagram and Lyapunov exponent of Logistic and Tent map. In Fig. 2.9, the bifurcation diagram and Lyapunov exponent are shown for ICM map and the proposed LT-ICM map. From Fig. 2.9, we can see that the generated map is distributed in maximum regions of phase space within $(0, r_3)$, which indicates the LT-ICM is highly unpredictable and has good ergodicity.

2.2.3.4 Lyapunov exponent(LE)

For a system to be chaotic it must have positive Lyapunov exponents. The value for proposed map is positive which tells the generated sequence is chaotic. The Logistic and Tent map have better chaotic behaviour in limited range. The ICM map has better chaotic range, but it has periodic behaviour for some range of controlling

parameter. The proposed chaotic map covers maximum range of controlling parameter over which lyapunov exponent is non-zero. Fig. 2.9 shows the distribution of LE with respect to the controlling parameter(r_3) within range(0,50). The Lyapunov Exponent for equation $x_{i+1} = G(x_i)$ is mathematically defined as follows:

$$\lambda_{G(x)} = \lim_{n\to\infty} (1/n) \ln \mid (G^n(x_0 + \epsilon) - G^n(x_0))/\epsilon \mid \tag{2.19}$$

where value of ϵ is very small. For higher control parameter, the ICM chaotic map behaves choatic in nature for entire range. In our proposed method, at lower value of control parameter there is no periodic window and it behaves more chaotic throughout the range of control parameter.

2.2.4 Proposed Encryption Scheme

2.2.4.1 Shot boundary detection

The first step for encrypting the video is to find the shots in the video. Prior to shot detection, all video frames are reshaped into a two-dimensional form by concatenating their R, G, and B components. As the input is color video, each frame of size $M \times N \times 3$ is transformed into a two-dimensional representation of size $M \times 3N$ by concatenating the R, G, and B channel matrices along the column direction. In the proposed method, this transformation follows a band-interleaved arrangement. Shot detection is performed by Sum Squared Difference(SSD) approach used in [57]. We first find the sum squared difference between two consecutive frames. If the total change is more than 75% then a shot boundary is detected at that frame [57].

2.2.4.2 Frame de-correlation within shot

The frames present in a shot contain temporal redundancy and it is removed by taking XOR difference between a frame and its previous frame. First a chaotic sequence of same size as each frame is generated using Algorithm-1. For the first frame of each shot, XOR difference is computed between itself and the chaotic sequence. For all other frames, XOR differences are computed from the current frame and its previous frame so that all the processed frames in each shot contain less numbers of non-zero

Algorithm 3 Pseudorandom sequence generation(PRSG) using LT-ICM

Input: $(D, x_0, a_0$, where D is frame of size [M,3N]
Output: Chaoticseq

Size of each frame $\leftarrow [M, 3N]$
Length of PRSG$\leftarrow [1, 3MN]$
$S = \sum_{i=1}^{3MN} D(i)$
if $S = 0$
$t \leftarrow 0$
else
$t = \frac{\sqrt{S}+5}{(S+255)}$
end
$x_1 = x_0 + t, a_1 = a_0 + t,;$
for $i = 1$ to $3MN$ **do**
    if $x_i < 0.5$ **then**
        $x_{i+1} = sin(a/(2ax_i + (1-a)x_i(1-x_i) + 0.5));$
    else
        $x_{i+1} = sin(a/(2a(1-x_i) + (1-a)x_i(1-x_i) + 0.5));$
    end if
$X(i) = floor(10^{10} \times (x_{i+1}))$ mod 256
end for
Chaoticseq=X;

values. Thresholding is applied on non-zero elements to eliminate non-significant difference values hence to reduce encryption time and memory usages.

2.2.4.3 Encryption of video

The sparse data obtained from the de-correlation stage are undergone chaotic sequence based permutation and substitution. Substituted, permuted and de-correlated data are converted into frame format, where unfilled positions in a frame are substituted with zero value. A new chaotic sequence is generated for every frame in each shot. The size of chaotic sequence is same as the dimension of each frame. The chaotic sequence generation process depends on initial state, controlling parameter and previous encrypted frame. Now XOR operation is performed between generated chaotic sequence and pixel values of current frame. All the encrypted frames in each shot are arranged to get color video frames in RGB form and encrypted shots are obtained after combining them. Finally all the encrypted shots are combined to get encrypted video. All the steps of video encryption are presented in Algorithm-4.

The decryption is the inverse of encryption process. Decryption is done by inverse substitution operation at frame level of each shot followed by inverse substitution and inverse permutation in sparse level and then inverse of de-correlation operation.

2.2.5 Experimental Results

We have tested our proposed method considering three standard videos used in [40], including Rhinos, Train, and Flamingo. We also used six different videos of 150 frames of various sizes, including Akiyo, Silent, Waterfall, Suzie, Salesman, and Highway, used in [59]. The experimental procedures were conducted on a desktop computer equipped with an Intel(R) Core(TM) i5-9300H processor operating at a clock speed of 2.40 GHz and paired with 8 GB of RAM. MATLAB 2020a was utilized for executing our scheme. We experimentally set $(x_0, a_0, x_1, a_1) = (0.65135, 30, 0.71835, 30.26180)$ as secret key. We compare our experimental results with different existing video encryption methods for different videos.

A) **Correlation Coefficient(CC) Analysis:** Ideally, the correlation coefficient of the original frame and the encrypted frame has to be 1 and 0 respectively. We perform experiment by considering 10000 randomly selected pixels with its adjacent pixels in all directions to find mean correlation coefficient. The mean CC for encrypted frame is nearer to zero for our method. Fig. 2.10 shows the distribution of adjacent pixels for frame number 1 of Akiyo video in all directions for each channel component.

B) **Histogram Analysis:** Histogram of encrypted video frames needs to have an uniform distribution for better security. Histogram of original frame number 1 of Akiyo video and its corresponding encrypted version for the R,G,B components are shown in Fig. 2.11. From Fig. 2.11, it is clear that our proposed method will be able to defend statistical attacks as encrypted frame has uniform histogram.

Algorithm 4 Encryption of video by divide-and-conquer approach

Input: (x_0, a_0, V), where V is input video
Output: Encrypted video

[r, c, d, N_1]=size(V) ▷ N_1 is the number of frames
Reshape all the frames of video(V) by concatenating R,G,B components and get new video V_1
Find the shot boundaries of V_1 using SSD
for $i = 1 : N_2$ **do** ▷ N_2 is the number of frames in the shot
  if $i = 1$ **then** ▷ De-correlating the frames in the shot
    $S_1 = PRSG(\mathbf{0}, x_0, a_0)$ ▷ Reshape S_1 of size same as of V_1
    $V_2(:,:,i) = V_1(:,:,i) \oplus S_1$
    else
    $V_2(:,:,i) = V_1(:,:,i-1) \oplus V_1(:,:,i)$
  end if
end for
Convert shot to sparse data by selecting only non-zero elements
[R,C,F,Val]=Shottosparse(shot)
Do thresholding by eliminating the pixel values within certain range e.g less than $5, 10, 15$ and 20
for i=1 to L **do** ▷ L is the length of non-zero elements in shot
  $S_2 = PRSG(\mathbf{0}, x_0, a_0)$
  $I_1 = sort(S_2)$ ▷ I_1 is the index of sorted S_2
  $Val_1(i) = Val(I_1(i))$
  if $i = 1$ **then**
    $Val_2(i) = Val_1(i) \oplus Val_1(L) \oplus S_2(i)$
    else
    $Val_2(i) = Val_2(i-1) \oplus Val_1(i) \oplus S_2(i)$
  end if
end for
Convert encrypted sparse data to dense form data from its indices and values
for $i = 1 : N_2$ **do**
  if $i = 1$ **then**
    $S_3 = PRSG(\mathbf{0}, x_1, a_1)$ ▷ Reshape S_3 of size same as of Val_2
    $Val_3(:,:,i) = Val_2(:,:,i) \oplus S_3$
    else
    $S_3 = PRSG(Val_3(:,:,i-1), x_1, a_1)$
    $Val_3(:,:,i) = Val_2(:,:,i) \oplus S_3$
  end if
end for
Reshape Val_3 to get color encrypted video frames and combine all encrypted frames to get encrypted shot
Repeat above steps for all the shots and combine all encrypted shots to get final encrypted video

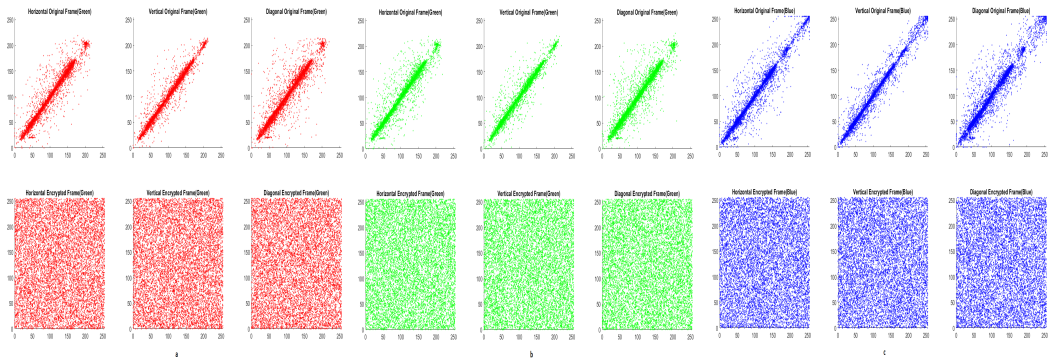


Figure 2.10: Distribution of adjacent pixels for frame no.1 of Akiyo video and its encrypted frame (a) Red, (b) Green, (c) Blue

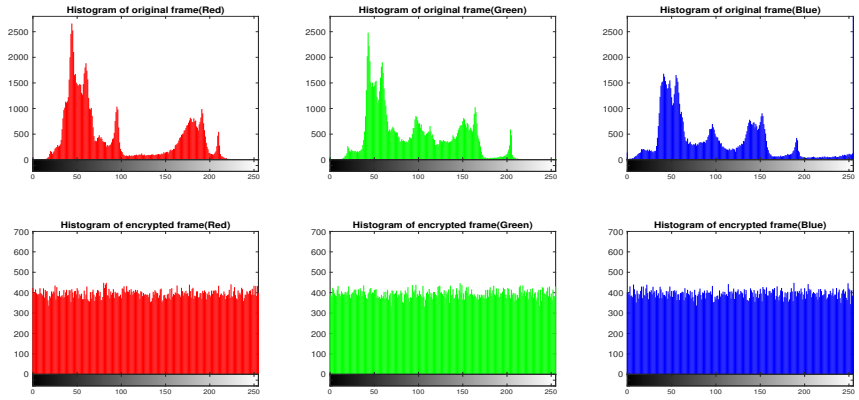


Figure 2.11: Histogram of Akiyo video for its Red, Green, Blue components of original frame no.1 and its encrypted frame

C) **Video Quality Analysis**:

The evaluation of video quality both after encryption and subsequent decryption is assessed through two metrics: the peak signal-to-noise ratio (PSNR) [62] and the structural similarity index (SSIM) [63]. To enhance security, the encrypted video must exhibit low PSNR and SSIM values. In every instance, the encrypted video's PSNR is below 9, underscoring the increased security of our method. The SSIM values are nearer to zero for all encrypted videos, which makes the videos secure. The PSNR and SSIM of decrypted video should be closer to those of input video for better perception. We have also tested all the videos for our cryptosystem by thresholding certain range of values in all the frames in shots. As the range of thresholding increases the mean encryption time decreases. However, the PSNR

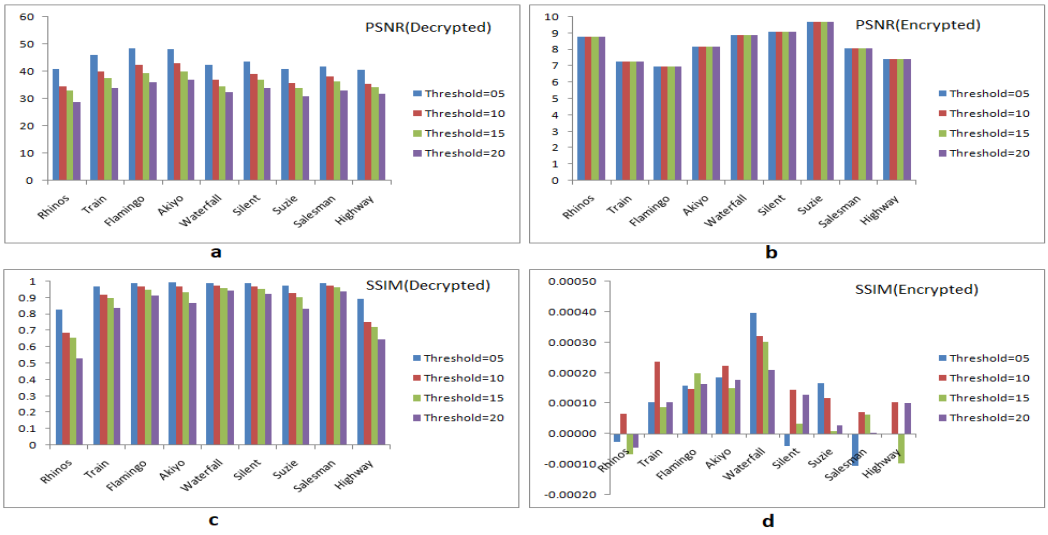


Figure 2.12: mean PSNR of decrypted and encrypted videos(a, b), mean SSIM of decrypted and encrypted videos

and SSIM of decrypted video decrease. So there is a trade-off between quality of the decrypted video and the mean encryption time. In Fig. 2.12, we have shown mean PSNR and mean SSIM of both encrypted and decrypted videos for different threshold values. For no thresholding, the mean PSNR and SSIM for decrypted videos are infinite and one, respectively.

D) **Comparison with External Approaches**: We compare results obtained by our proposed method with other video/image encryption methods in terms of key space, mean encryption time, entropy and mean correlation coefficient. Entropy is used to measure the randomness of encrypted frames and mean correlation coefficient is used to measure correlation between adjacent pixels.

We now provide a comparison of **mean execution times**. We consider different cases for encrypting the video like taking all non-zero pixel values in processed frames in each shot and eliminating the pixels having values less than 5,10,15 and 20. It take less time compared to other method for all cases. However, it take very less time if we eliminate the processed pixel values less than 5,10,15 and 20 with the loss of some quality in the video. So the proposed method for encrypting the video is very efficient with respect to memory usage and real-time security application. Table 2.8 shows the comparison of mean encryption time for different methods.

Table 2.8: Comparison of mean encryption time in seconds

Video names	Proposed method with thresholding					Valli et al. (2017) [40]			Ranjithkumar et al. (2019) [52]	Sethi et al. (2021) [58]
	No	5	10	15	20	Ikeda DDE	12D	8D		
Rhinos	0.6533	0.5421	0.4883	0.3911	0.4043	1.2147	1.7964	0.9124	0.5403	0.7946
Train	0.4485	0.3770	0.3313	0.2963	0.2857	0.9908	1.6493	0.8631	0.5178	0.8548
Flamingo	0.4228	0.3635	0.3030	0.2717	0.2555	1.1124	0.8402	0.8062	0.5982	0.8538

We now present a **key space analysis**. For any encryption technique 2^{128} is a reasonable key space such that it can resist brute force attack. Our method consists of four parameters in secret key (x_0, a_0, x_1, a_1). The precision for each parameter is considered as 10^{-14}, so the total key space is $10^{14\times4} = 10^{56} = 2^{186}$. So our method can defend against exhaustive brute force attack. Table 2.9 shows the comparison of key space with some recent methods. Decryption is not possible if there is very minor change in secret key which makes our system more robust against differential attacks. A minor change in secret key will produce different decrypted frame than that of original one which has been shown in Fig. 2.13. The attacker has knowledge of both plaintext and its ciphertext in case of known plaintext attack. In case of chosen plaintext attack, the attacker can get encrypted frames if some portion of input frames are imparted. As the chaotic sequence generation depends on substituted previous frames in the shot, so any minor change in secret key can make decryption impossible. Hence, our proposed method can defend both known and chosen plaintext attacks.

Table 2.9: Comparison of key space

Methods	El-Latif et al. (2020) [59]	Valli et al. (2017) [40]			Ranjithkumar et al. (2019) [52]	Kezia et al. (2008) [42]	Sethi et al. (2021) [58]	Proposed method
		8D	12D	Ikeda DDE				
Keyspace	2^{305}	8×2^{128}	3072×2^{128}	49152×2^{128}	2^{212}	2^{128}	256×2^{256}	2^{186}

Since the proposed scheme incorporates plaintext-dependent chaotic sequence generation, even a minor change in the plaintext leads to a substantially different chaotic sequence and hence a completely different decryption result. To quantify this behavior, Table 2.10 presents a comparison between the decrypted video frames obtained under plaintext perturbation and the corresponding original video frames.

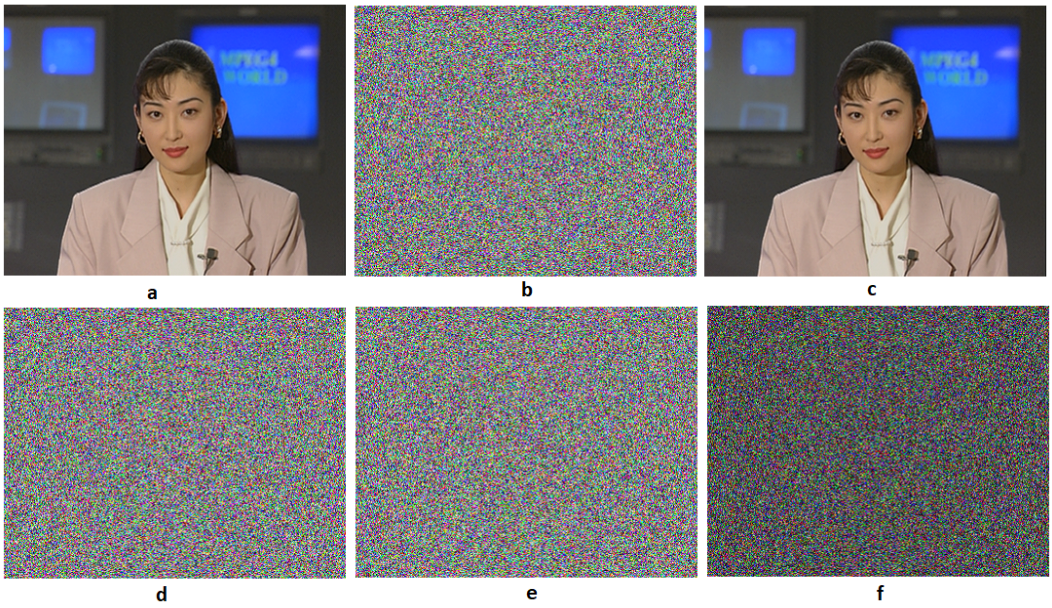


Figure 2.13: (a) Original first frame of Akiyo video (b) encryption with key 0.65135, 30, 0.71835, 30.26180(c) decryption with key 0.65135, 30, 0.71835, 30.26180(d) decryption with key $0.65135+10^{-14}$, 30, 0.71835, 30.26180 (e) decryption with key $0.65135,30+10^{-14}$, 0.71835, 30.26180 (f) result of d - e

It can be observed that the decrypted frames produce very low PSNR values (less than 10 dB) and SSIM values close to zero, which indicate severe degradation and almost no structural resemblance to the original frames. These results confirm that the proposed scheme is highly sensitive to plaintext changes and is therefore robust against plaintext-based attacks, making it extremely difficult for an attacker to recover meaningful visual information.

Table 2.10: Quantitative comparison between decrypted and original video

Video	**PSNR (dB)**	**SSIM**
Rhinos	9.6393	0.0126
Flamingo	8.3249	0.0133
Train	9.0045	0.0681

Finally, Table 2.11 shows the comparison of mean correlation coefficient and entropy of encrypted videos for different schemes.

From the above comparisons, it becomes evident that our method achieves good performance in terms of mean execution time, keyspace analysis, mean correlation coefficient and entropy values.

Table 2.11: Comparison of average CC and Entropy

Methods	CC after Encryption			Entropy
	Horizontal	Vertical	Diagonal	
El-Latif et al. (2020) [59]	-0.0012	0.0004	0.0003	7.9984
Song et al. (2020) [60]	-0.0245	-0.0135	-0.0143	7.8708
Muhammad et al. (2018) [49]	0.0015	-0.0007	0.0006	**7.9998**
Unde et al. (2020) [34]	0.0030	0.0027	0.0014	7.9900
Sethi et al. (2021) [58]	0.0020	0.0048	0.0114	7.9993
Our proposed method	**-0.0002**	**-0.0002**	**-0.0002**	7.9989

2.2.6 Discussion

In this work, we proposed a fast and secure video encryption method. A divide-and-conquer approach is employed to make the encryption fast by processing frames at shot level. Each shot is de-correlated and only pixels above a certain threshold within the difference frames are further processed. Secure sort based random permutation is used for shuffling and resulting shuffled data in each shot is substituted by frame independent chaotic sequence. The processed pixel values are converted into frame level. Once again the frames in each shot are substituted by a frame dependent unique sequence generated by the proposed LT-ICM chaotic map. Experimental results demonstrate that the proposed method has good result over existing approaches in terms of mean encryption time, mean correlation coefficient, key sensitivity, entropy, mean PSNR and SSIM.

Chapter 3

Compressive Video Sensing for High-Quality Video Recovery

In this chapter, we introduce a deep learning-based video Compressive Sensing (CS) method for high-quality video recovery, that can be useful for diverse applications, like telemedicine and cloud-based surveillance. We split a video into a number of Groups Of Pictures (GOPs) with each GOP consisting of both keyframes and non-keyframes. The proposed video CS method uses a convolutional neural network (CNN) with Structural Similarity Index Measure (SSIM) based loss function. In a GOP, a convolution layer performs block-based frame-wise sampling, which optimizes the sample matrix. The recovery process has two stages. In the initial stage, CNN is employed to make efficient use of spatial redundancy. In the deep recovery stage, non-keyframes are compensated utilizing both keyframes and neighboring non-keyframes using multilevel feature compensation and single-level feature compensation, respectively. Through extensive experimentation, we establish the efficacy of our solution over several state-of-the-art image and video CS methods.

3.1 Introduction

In recent years, video compression schemes have received considerable attention. We have addressed video compression to achieve high-quality video recovery with low computational costs. Our solution can be helpful for practical applications like telemedicine [64] and cloud-based surveillance [65]. Video compression with different

video codecs [66, 67] has complex encoder and the encoder requires more computational power for processing the video data than the decoder. So, such schemes are not well suited for low-power applications. On the other hand, the CS concept which deals with sampling and representing a video signal with a few non-zero coefficients is required to form a simple encoder. In compressive video sensing (CVS), compression and sampling are done simultaneously which is desirable for many imaging applications [68, 69]. Two methods [70, 71] use image-based CS while recovering the compressed frames. These methods treat each video frame as an image. Image-based CS methods provide low-quality recovered video frames because they ignore temporal correlation. Different CVS methods [72, 73] use both spatial and temporal correlation while recovering the complete video by considering the whole video sequence as a volume. However, these methods have high computational complexity [69]. Most CVS methods ([24, 74, 75]) perform frame-wise initial reconstruction using the measurements and then perform motion estimation and compensation to improve current frame reconstruction. Investigating temporal correlation is the key challenge while applying deep learning-based image CS to video CS. In [20], authors proposed a video CS method where they compensate non-keyframes using multilevel feature compensation with the help of only keyframes for achieving better quality frames. To further improve the quality of the recovered video frames, we consider compensation from keyframes as well as neighboring non-keyframes.

In this work, we present a video compression scheme at the GOP level where compression is performed using deep CS. The sampling matrix is optimized using the convolution layer during video CS sampling. The recovery procedure is divided into two stages, namely, frame-wise initial recovery and multilevel feature compensation-based deep recovery. Non-keyframes are compensated from both keyframes in multilevel and neighboring non-keyframes in single level to ascertain good quality of the recovered video. In this work, instead of the traditional CVS, we use a deep learning-based CVS. Secondly, we optimize the measurement matrix using the convolution layer as opposed to a content-independent measurement matrix. Thirdly, we have shown here how the quality of the recovered video has improved at lower

sampling ratios (SR). Finally, we have incorporated extensive experimentation using six standard CIF video sequences. Our key contributions are now summarized below:

1. We propose a GOP level scheme for video compression. The proposed scheme recovers good quality video and exhibits reasonable recovery time compared to traditional CVS schemes.

2. Our SSIM based deep CS scheme compensates non-keyframes in a GOP from both keyframes and neighboring non-keyframes. This method recovers high quality videos even though compression is performed at very low sampling ratios.

3. Our proposed scheme has two recovery mechanisms which has comparable lower recovery time with high recovery quality. In single keyframe-based recovery network, one keyframe and nearby non-keyframes compensate the non-keyframes in a GOP, while double keyframe-based network uses two keyframes to offer bidirectional compensation for the non-keyframes as well as compensation from neighboring non-keyframes.

The rest of the work is organized as follows: In Section 3.2, we discuss the related works and point out their limitations. In Section 3.3, we describe our proposed video compression in detail. Section 3.4 presents the experimental results with analysis. Finally, the work has a summary in Section 3.5.

3.2 Related Work

Recent deep learning-based image CS approaches achieve high-quality recovery [29–31, 76, 77]. In [76], authors developed a structured deep network called ISTA-Net, which was influenced by a technique called iterative shrinkage thresholding. The goal was to optimize a general l_1 norm based CS reconstruction model. Using a basic CNN model, termed as ReconNet, Kulkarni *et al.* [29] showed that noniterative recovery was possible. Shi *et al.* [30, 31] came up with the idea of using a CNN to train

both the sampling matrix and the reconstruction network jointly. Authors in [20] implemented video compression using different image-based CS methods [29], [30,31]. However, any image-based CS method applied for video compression invariably omits temporal information across the frames, resulting in a low-quality recovered video.

Now, we discuss some CS-based video compression techniques using multiple hypothesis (MH) prediction paradigms. Using MH predictions of the current frame, the authors in [75] suggested a method to recover the video. This method generates a residual in the compressed sensing random projection domain. In [74], authors proposed a two-phase video CS recovery scheme using reweighted residual sparsity (RRS). Even though the video quality is better with RRS, it takes more time for the recovery of video. Another MH-based method [24] considered motion estimation and motion compensation residuals to recover video frames and their associated motion fields. The video recovery process takes more time when MH-based methods([24], MH-ST [25],MH-RTIK [26]) are used. Further, many MH-based methods [74], [75], [26], [25] are not suitable for very low sampling ratios like 0.01 and 0.05 because it is extremely difficult to recover high-quality video with a handful of samples. Shi et. al [20] proposed an end-to-end DL-based video CS scheme known as video compressive sensing network (VCSNet) using convolutional neural network (CNN) to improve video quality. In their scheme, non-keyframes are sampled at a lower rate and compensated using keyframes that are sampled at a higher rate. They utilized a loss function based on mean square error (MSE), which performs substantially worse than our loss function based on SSIM.

3.3 Proposed method

3.3.1 Single Keyframe based Deep Network

We divide the whole network into 3 stages, namely, frame-wise sampling or block-level compressive sensing (BCS), initial recovery (frame-wise), and deep recovery. A multilevel feature compensation approach is employed where the current non-keyframes are compensated using both keyframe and neighboring non-keyframes in

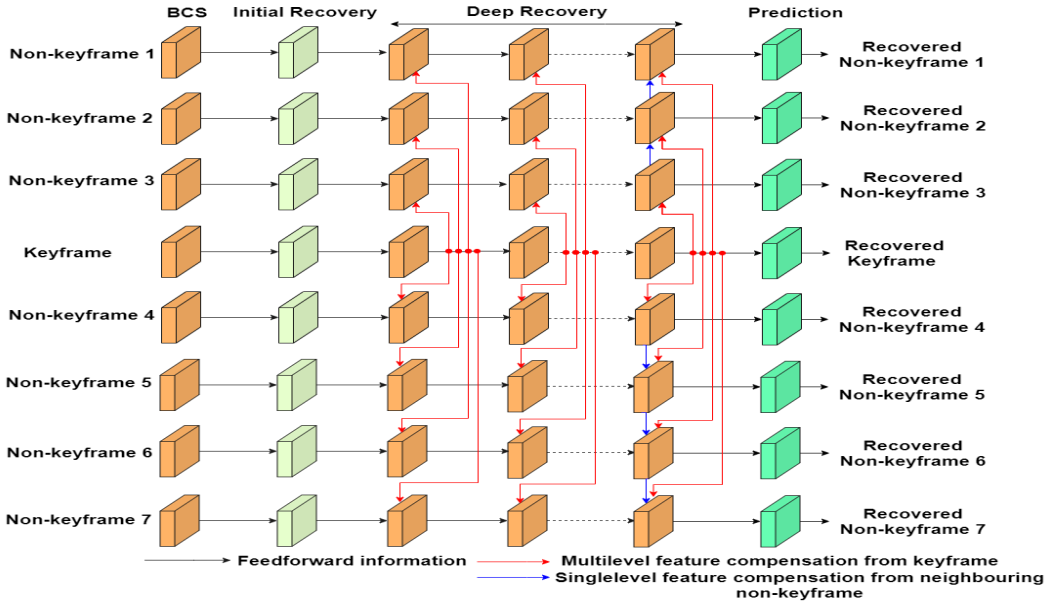


Figure 3.1: Architecture of proposed CVS with single keyframe

the deep recovery stage. We divide the video into GOPs of 8 frames each. The keyframe is a representative frame that is sampled and recovered independently. The first frame of GOP could be designated as a keyframe [20]. However, when there is more motion and a quick shift in video frames inside a GOP, the initial frame may not necessarily include important information. Choosing other frames as keyframes in a GOP might better fit situations like content variation, large motion, and scene change. In a single keyframe-based deep network, we have considered the fourth frame of each GOP as the keyframe since it is in the center of the GOP and used to compensate non-keyframes equally in both directions. In Fig. 3.1, we show the recovery of video frames using our single keyframe-based CVS method.

3.3.1.1 Block-Based Sampling (Frame-wise)

Block sampling in a frame-wise approach is preferred over sampling complete sets of spatial and temporal information at one shot due to lower complexity. Let us consider a block of size $B \times B$, number of channels l (for this work, $l = 1$ as we deal with only grayscale video) and a sampling ratio s. Then, block-based frame-wise

sampling can be represented by:

$$y = \Phi \times x \tag{3.1}$$

Here, $\Phi \in \mathbb{R}^{sB^2 \times B^2}$ is the measurement matrix, $x \in \mathbb{R}^{B^2 \times 1}$ is a column vector representing a block, and $y \in \mathbb{R}^{sB^2 \times 1}$ is another column vector representing the measurement data of the same block. Note that Φ can be replaced with a convolutional kernel with special size and stride for key and non-keyframes. A frame is divided into $m \times n$ blocks where each block has a size of $B \times B$, and every row of the measurement matrix is replaced by a $B \times B$ kernel of the convolution filter. One compressed measurement is obtained by convolving one $B \times B$ kernel across one block. So, to get sB^2 compressed features from one block we need sB^2 kernels. The whole frame is convolved using non-overlapping blocks with different numbers of kernels by taking a stride of $B \times B$. Compressive sensing of a whole video frame obtained through convolution can be mathematically expressed as:

$$\mathbf{Y}_{\mathrm{K}} = \mathbf{W}_{\mathrm{K}} * \mathbf{X}_{\mathrm{K}} \tag{3.2}$$

$$\mathbf{Y}_{\mathrm{N}} = \mathbf{W}_{\mathrm{N}} * \mathbf{X}_{\mathrm{N}} \tag{3.3}$$

where $\mathbf{Y}_{\mathrm{K}}$, $\mathbf{X}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{K}}$ denote the measurement data, frame-level data, weights of the convolutional network for keyframe (K) respectively. Similarly, $\mathbf{Y}_{\mathrm{N}}$, $\mathbf{X}_{\mathrm{N}}$, and $\mathbf{W}_{\mathrm{N}}$ denote the measurement data, frame-level data, and weights of the convolutional network for the N^{th} non-keyframe respectively. The keyframes are sampled at a higher sampling ratio ($s^{'}$) to compensate for non-keyframes that are sampled at a lower sampling ratio ($s^{''}$). Note that the number of filters used in weights depends on the sampling ratio of compression. So, $\mathbf{W}_{\mathrm{K}}$ has $s^{'} \times B^2$ number of filters and $\mathbf{W}_{\mathrm{N}}$ has $s^{''} \times B^2$ number of filters. The fourth frame of a GOP is the keyframe, whereas the rest of the frames are non-keyframes.

3.3.1.2 Initial recovery (Frame-wise)

Since block-by-block reconstruction may give rise to blocking artifacts it is preferable to optimize the images as a whole frame instead of blocks [20]. Thus, this stage deals with the decompression of compressed measurement blocks followed by reshaping and concatenation of the blocks to convert the measurement data into frame-level data. The measurement data y for each block of size $1 \times 1 \times sB^2$ is decompressed or de-convoluted into size $1 \times 1 \times B^2$, which requires B^2 number of filters of size $1 \times 1 \times sB^2$. So, we write:

$$\hat{\mathbf{X}}_{\mathrm{K}} = \mathbf{W}_{\mathrm{K}}^{\mathrm{ir}} * \mathbf{Y}_{\mathrm{K}} \tag{3.4}$$

$$\hat{\mathbf{X}}_{\mathrm{N}} = \mathbf{W}_{\mathrm{N}}^{\mathrm{ir}} * \mathbf{Y}_{\mathrm{N}} \tag{3.5}$$

where $\hat{\mathbf{X}}_{\mathrm{K}}$ and $\hat{\mathbf{X}}_{\mathrm{N}}$ denote the initial recovery of the block data and $\mathbf{W}_{\mathrm{K}}^{\mathrm{ir}}$ and $\mathbf{W}_{\mathrm{N}}^{\mathrm{ir}}$ denote weights of the preliminary recovery stage. $\mathbf{W}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{N}}$ have support of $1 \times 1 \times s^{'}B^2$ and $1 \times 1 \times s^{''}B^2$ respectively. The recovered block-level data are in vector form of size B^2.

Using the recovered blocks from the compressive measurements, the initial recovery of each frame is performed by placing every block back into its original spatial position and concatenating them row-wise and column-wise to reconstruct the complete frame for both keyframes and non-keyframes.

3.3.1.3 Deep recovery

During this phase, a deep recovery network is used to enhance the quality of recovered frames. Keyframe and non-keyframe measurements are typically temporally associated within a GOP, and keyframe measurements contain more information than non-keyframe measurements.As a result, the features obtained from the deep recovery of keyframes are used to compensate for the non-keyframes. Also, neighboring frames are highly correlated, which are also utilized to compensate for the

next frame. This recovery of keyframe may be expressed mathematically as [20] :

$$\mathbf{D}_{\mathrm{K}}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) = \mathrm{Relu}\left(\mathbf{W}_{\mathrm{K}}^{1} * \hat{\mathbf{X}}_{\mathrm{K}} + \mathbf{b}_{\mathrm{K}}^{1}\right) \tag{3.6}$$

$$\mathbf{D}_{\mathrm{K}}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) = \mathrm{Relu}\left(\mathbf{W}_{\mathrm{K}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{K}}^{\mathrm{i}-1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{\mathrm{i}}\right) \tag{3.7}$$

$$\mathrm{i} = 2, 3 \cdots, n-1$$

$$\mathbf{D}_{\mathrm{K}}^{\mathrm{n}}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) = \mathbf{W}_{\mathrm{K}}^{\mathrm{n}} * \mathbf{D}_{\mathrm{K}}^{\mathrm{n}-1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{\mathrm{n}} \tag{3.8}$$

where Relu is the activation function and $\mathbf{D}_{\mathrm{K}}$ is a deep network that cascades many convolution layers and activation layers. The first stage of the deep reconstruction network $\mathbf{W}_{\mathrm{K}}^{1}$ has d filters of size $f \times f \times l$. The later stages of the network (denoted by $i = 2, 3, \cdots, n-1$) consist of d filters of size $f \times f \times d$ as there are d features exploited for spatial feature compensation. In the final stage $\mathbf{W}_{\mathrm{K}}^{\mathrm{n}}$ comprises l filters of size $f \times f \times d$. The term b represents bias for the respective layers. In particular, size of $\mathbf{b}_{\mathrm{K}}^{\mathrm{i}}$ for $(i = 1, 2, \cdots, n-1)$ and $\mathbf{b}_{\mathrm{K}}^{\mathrm{n}}$ are $d \times 1$ and $l \times 1$ respectively. The non-keyframe network follows the same feed-forward structure as the keyframe during the deep recovery network, but it additionally has multilevel feature compensation from the keyframe. We also include the multilevel feature compensation from the neighboring non-keyframes while compensating the current non-keyframe in the pre-final deep reconstruction block to improve the quality of the recovered video. The non-keyframe closest to the keyframe is recovered using the multilayer feature compensation received only from the keyframe. Except for the frames closest to the keyframe, the network equation for each non-keyframe may be written as follows:

$$\mathbf{D}_{\mathrm{N}}^{1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) = \mathrm{Relu}\left(\ \mathbf{W}_{\mathrm{N}}^{1} * \hat{\mathbf{X}}_{\mathrm{N}} + \mathbf{b}_{\mathrm{N}}^{1} + \mathbf{W}_{\mathrm{K}}^{1} * \mathbf{D}_{\mathrm{K}}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{1}\ \right) \tag{3.9}$$

$$\mathbf{D}_{\mathrm{N}}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) = \mathrm{Relu}\left(\ \mathbf{W}_{\mathrm{N}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{N}}^{\mathrm{i}-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) + \mathbf{b}_{\mathrm{N}}^{\mathrm{i}} + \mathbf{W}_{\mathrm{K}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{K}}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{\mathrm{i}}\ \right) \tag{3.10}$$

$$\mathrm{i} = 2, 3 \cdots, n-2$$

$$\mathbf{D}_{\mathrm{N}}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) = \mathrm{Relu}\left(\begin{array}{l} \mathbf{W}_{\mathrm{N}}^{n-1} * \mathbf{D}_{\mathrm{N}}^{n-2}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) + \mathbf{b}_{\mathrm{N}}^{n-1} + \mathbf{W}_{\mathrm{K}}^{n-1} * \mathbf{D}_{\mathrm{K}}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{n-1} \\ + \mathbf{W}_{\mathrm{N}-1}^{n-1} * \mathbf{D}_{\mathrm{N}-1}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}-1}\right) + \mathbf{b}_{\mathrm{N}-1}^{n-1} + \mathbf{W}_{\mathrm{N}+1}^{n-1} * \mathbf{D}_{\mathrm{N}+1}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}+1}\right) \\ + \mathbf{b}_{\mathrm{N}+1}^{n-1} \end{array}\right) \tag{3.11}$$

$$\mathbf{D}_{\mathrm{N}}^{\mathrm{n}}\left(\hat{\mathbf{X}}_{N}\right) = \mathbf{W}_{\mathrm{N}}^{\mathrm{n}} * \mathbf{D}_{\mathrm{N}}^{\mathrm{n}-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) + \mathbf{b}_{\mathrm{N}}^{\mathrm{n}} \tag{3.12}$$

where $\mathbf{W}_{\mathrm{N}}^{1}$ and $(\mathbf{W}_{\mathrm{K}}^{1}, \mathbf{W}_{\mathrm{N}}^{\mathrm{i}}, \mathbf{W}_{\mathrm{K}}^{\mathrm{i}})$ has d filters of size $f \times f \times l$ and $f \times f \times d$ respectively. The different biases $\mathbf{b}_{\mathrm{N}}^{1}$, $\mathbf{b}_{\mathrm{K}}^{1}$, $\mathbf{b}_{\mathrm{N}}^{\mathrm{i}}$, $\mathbf{b}_{\mathrm{K}}^{\mathrm{i}}$, $\mathbf{b}_{\mathrm{N}}^{n-1}$, $\mathbf{b}_{\mathrm{K}}^{n-1}$ and $\mathbf{b}_{\mathrm{N}\pm1}^{n-1}$ have size of $d \times 1$. $\mathbf{W}_{\mathrm{N}}^{\mathrm{n}}$ has l filters of size $f \times f \times d$ and $\mathbf{b}_{\mathrm{N}}^{\mathrm{n}}$ has size of $l \times 1$. The spatial and temporal redundan-

cies are fused with the help of deep multilevel feature compensation using Eq.(3.9), Eq.(3.10), and Eq.(3.11). $\mathbf{W}_{\mathrm{N}}^{1} * \hat{\mathbf{X}}_{\mathrm{N}} + \mathbf{b}_{\mathrm{N}}^{1}$ in Eq.(3.9) and $\mathbf{W}_{\mathrm{N}}^{i} * \mathbf{D}_{\mathrm{N}}^{\mathrm{i}-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right) + \mathbf{b}_{\mathrm{N}}^{\mathrm{i}}$ in Eq.(3.10) exploit spatial/intraframe redundancy while $\mathbf{W}_{\mathrm{K}}^{1} * \mathbf{D}_{\mathrm{K}}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{1}$ in Eq.(3.9) and $\mathbf{W}_{\mathrm{K}}^{i} * \mathbf{D}_{\mathrm{K}}^{i}\left(\hat{\mathbf{X}}_{\mathrm{K}}\right) + \mathbf{b}_{\mathrm{K}}^{i}$ in Eq.(3.10) exploit temporal/interframe redundancy. Eq.(3.11) further indicates that the recovery of non-keyframes is guided not only by the keyframes but also by adjacent non-keyframes, thereby enabling better utilization of temporal correlation. Eq. (3.12) describes the fusion of intraframe and interframe information to obtain the final recovered video frame.

3.3.2 Double Keyframe based Deep Network

As the GOP size increases, the temporal distance between keyframes and some non-keyframes becomes larger, leading to reduced inter-frame redundancy. Consequently, non-keyframes that are farther from the reference keyframe cannot be effectively compensated using information derived from a single keyframe. To solve this issue, we suggest compensating non-keyframes using double keyframes along with nearer non-keyframes. Consequently, the non-keyframe recovery acquires compensation from i) both forward and backward frames and ii) nearer non-keyframes. We divide video into GOPs, each of which has 8 frames. With double keyframe-based CVS, the first frames of two successive GOPs are considered a keyframe while recovering all frames for the specific GOP [20]. We choose one keyframe from the current GOP and the second keyframe from the next GOP rather than two keyframes from the same GOP. This reduces the number of samples in the GOP after BCS.

Figure 3.2 shows the GOP structure for our CVS system based on a double keyframe. We have marked F_1 as keyframe 1, which is the first frame of the first GOP, and F_9 as keyframe 2, which is the first frame of the second GOP; all other frames between these two keyframes (F_2 to F_8) are designated as non-keyframes. During reconstruction, each non-keyframe (F_2 to F_8) within the current GOP is jointly compensated using features from both keyframe 1 and keyframe 2, along with adjacent non-keyframes. In Fig. 3.3, we show the recovery of video frames using our double keyframe-based CVS method.

Our proposed double keyframe network uses block-based frame-wise sampling, frame-wise initial recovery, and deep recovery with multi-level feature compensation. Our double keyframe-based network's frame-wise initial recovery and block-based frame-wise sampling are similar to those of our single keyframe-based network. The sampling of the second keyframe and the initial recovery of that frame is the ad-

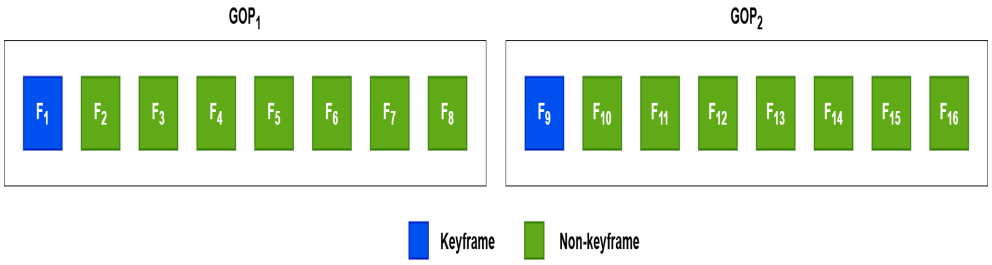


Figure 3.2: GOP structure of proposed CVS with double keyframe

ditional processes for the second keyframe. Second keyframe sampling and initial recovery may be formulated by Eq.(3.2)-(3.5).

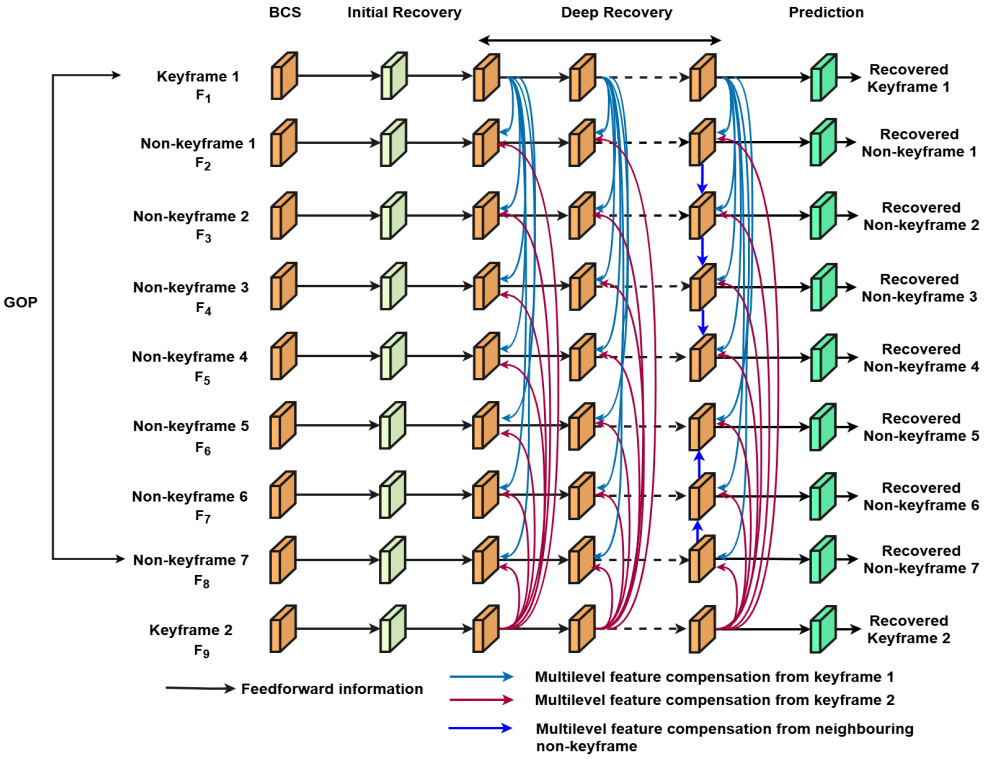


Figure 3.3: Architecture of proposed CVS with double keyframe

The non-keyframe network follows the same feed-forward structure of the keyframe during the deep recovery network but it additionally has multilevel feature compensation from the double keyframes and neighboring non-keyframes. The non-keyframe nearest to the keyframe is compensated using only keyframes. In all stages of the deep recovery network except the $(n-1)^{th}$, two keyframes are used to compensate the N^{th} non-keyframe. The N^{th} non-keyframe at different stages of the deep recovery network is recovered using multilevel feature compensation from two

keyframes, and neighboring non-keyframes which can be written as follows:

$$\mathbf{D}_{\mathrm{N}}^{1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)=\mathrm{Relu}\left(\begin{array}{l}\mathbf{W}_{\mathrm{N}}^{1} * \hat{\mathbf{X}}_{\mathrm{N}}+\mathbf{b}_{\mathrm{N}}^{1}+\mathbf{W}_{\mathrm{K1}}^{1} * \mathbf{D}_{\mathrm{K},1}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K},1}\right)+\mathbf{b}_{\mathrm{K1}}^{1} \\ +\mathbf{W}_{\mathrm{K2}}^{1} * \mathbf{D}_{\mathrm{K},2}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K},2}\right)+\mathbf{b}_{\mathrm{K2}}^{1}\end{array}\right) \tag{3.13}$$

$$\mathbf{D}_{\mathrm{N}}^{i}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)=\mathrm{Relu}\left(\begin{array}{l}\mathbf{W}_{\mathrm{N}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{N}}^{\mathrm{i}-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)+\mathbf{b}_{\mathrm{N}}^{\mathrm{i}}+\mathbf{W}_{\mathrm{K1}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{K},1}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{K},1}\right)+\mathbf{b}_{\mathrm{K1}}^{\mathrm{i}} \\ +\mathbf{W}_{\mathrm{K2}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{K},2}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{K},2}\right)+\mathbf{b}_{\mathrm{K2}}^{\mathrm{i}}\end{array}\right) \tag{3.14}$$

$$i=2,3 \cdots, n-2$$

$$\mathbf{D}_{\mathrm{N}}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)=\mathrm{Relu}\left(\begin{array}{l}\mathbf{W}_{\mathrm{N}}^{n-1} * \mathbf{D}_{\mathrm{N}}^{n-2}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)+\mathbf{b}_{\mathrm{N}}^{n-1}+\mathbf{W}_{\mathrm{K1}}^{n-1} * \mathbf{D}_{\mathrm{K},1}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{K},1}\right)+\mathbf{b}_{\mathrm{K1}}^{n-1} \\ +\mathbf{W}_{\mathrm{K2}}^{n-1} * \mathbf{D}_{\mathrm{K},2}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{K},2}\right)+\mathbf{b}_{\mathrm{K2}}^{n-1}+\mathbf{W}_{\mathrm{N}-1}^{n-1} * \mathbf{D}_{\mathrm{N}-1}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}-1}\right) \\ +\mathbf{b}_{\mathrm{N}-1}^{n-1}+\mathbf{W}_{\mathrm{N}+1}^{n-1} * \mathbf{D}_{\mathrm{N}+1}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}+1}\right)+\mathbf{b}_{\mathrm{N}+1}^{n-1}\end{array}\right) \tag{3.15}$$

$$\mathbf{D}_{\mathrm{N}}^{n}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)=\mathbf{W}_{\mathrm{N}}^{\mathrm{n}} * \mathbf{D}_{N}^{n-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)+\mathbf{b}_{\mathrm{N}}^{\mathrm{n}} \tag{3.16}$$

The expression $\mathbf{W}_{\mathrm{N}}^{1} * \hat{\mathbf{X}}_{\mathrm{N}}+\mathbf{b}_{\mathrm{N}}^{1}$ in Eq.(3.13) and $\mathbf{W}_{\mathrm{N}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{N}}^{\mathrm{i}-1}\left(\hat{\mathbf{X}}_{\mathrm{N}}\right)+\mathbf{b}_{\mathrm{N}}^{\mathrm{i}}$ in Eq.(3.14) indicate spatial correlated data. $\mathbf{W}_{\mathrm{K1}}^{1} * \mathbf{D}_{\mathrm{K},1}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K},1}\right)+\mathbf{b}_{\mathrm{K1}}^{1}$ in Eq.(3.13) and $\mathbf{W}_{\mathrm{K1}}^{i} * \mathbf{D}_{\mathrm{K},1}^{\mathrm{i}}\left(\hat{\mathbf{X}}_{\mathrm{K},1}\right)+\mathbf{b}_{\mathrm{K1}}^{\mathrm{i}}$ in Eq.(3.14) show the compensation data from 1^{st} keyframe. $\mathbf{W}_{\mathrm{K2}}^{1} * \mathbf{D}_{\mathrm{K},2}^{1}\left(\hat{\mathbf{X}}_{\mathrm{K},2}\right)+\mathbf{b}_{\mathrm{K2}}^{1}$ in Eq.(3.13) and $\mathbf{W}_{\mathrm{K2}}^{\mathrm{i}} * \mathbf{D}_{\mathrm{K},2}^{i}\left(\hat{\mathbf{X}}_{\mathrm{K},2}\right)+\mathbf{b}_{\mathrm{K2}}^{\mathrm{i}}$ in Eq.(3.14) represent compensation data from 2^{nd} keyframe. The fusion of spatial and temporal correlation utilizing multilevel feature compensation is expressed in Eq.(3.15). The final video frame is recovered by combining spatial and temporal data using Eq.(3.16).

3.3.3 Training

To optimize the network for non-keyframes, predicted output of the network and initial frame-level restoration are considered in the loss function. For the keyframe, only the final prediction of deep reconstruction is considered in the loss function. Let there be one keyframe and P number of non-keyframes in a GOP. As a result, the total number of loss functions for the single keyframe-based network is 2P+1. Likewise, the total number of loss functions for a double keyframe will be 2P+2. The loss function based on the structural similarity index (SSIM) [78] works substantially better than the Euclidean loss function or the mean square error (MSE) when it comes to image restoration. So we employ the SSIM-based loss function

in our proposed method. For a single keyframe-based network, the loss function is expressed as follows:

$$\mathbf{ESSIM}\left(\mathbf{X}_{\mathrm{K}}, \mathbf{D}_{\mathrm{K}}\right) = \mathbf{1} - \frac{\mathbf{1} + \mathbf{SSIM}\left(\mathbf{X}_{\mathrm{K}}, \mathbf{D}_{\mathrm{K}}\right)}{\mathbf{2}} \tag{3.17}$$

where ESSIM is the SSIM error found by calculating structural dissimilarities between the restored frame ($\mathbf{D}_{\mathrm{K}}$) and the original frame ($\mathbf{X}_{\mathrm{K}}$). Hence, the total loss for training M GOPs consisting of P non-keyframes and a single keyframe can be expressed as:

$$l\left(\theta\right) = \sum_{\mathrm{i}=1}^{\mathrm{M}} \left(\begin{array}{l} \left|\mathbf{ESSIM}\left(\mathbf{X}_{\mathrm{K}}^{(\mathrm{i})}, \mathbf{D}_{\mathrm{K}}^{(\mathrm{i})}\right)\right| + \\ \sum\limits_{\mathrm{N}=1}^{\mathrm{P}} \left(\left|\mathbf{ESSIM}\left(\mathbf{X}_{\mathrm{N}}^{(\mathrm{i})}, \mathbf{D}_{\mathrm{N}}^{(\mathrm{i})}\right)\right| + \left|\mathbf{ESSIM}\left(\mathbf{X}_{\mathrm{N}}^{(\mathrm{i})}, \hat{\mathbf{X}}_{\mathrm{N}}^{(\mathrm{i})}\right)\right| \right) \end{array} \right) \tag{3.18}$$

where θ is the parameter set, $\mathbf{D}_{\mathrm{K}}^{(\mathrm{i})}$ and $\mathbf{D}_{\mathrm{N}}^{(\mathrm{i})}$ are the deep recovery of keyframe $\mathbf{X}_{\mathrm{K}}^{(\mathrm{i})}$ and non-keyframe $\mathbf{X}_{\mathrm{N}}^{(\mathrm{i})}$ respectively. Similarly, $\hat{\mathbf{X}}_{\mathrm{N}}^{(\mathrm{i})}$ is the initial recovery of non-keyframe $\mathbf{X}_{\mathrm{N}}^{(\mathrm{i})}$.

We pretrained our model using weights of network [20]. We kept the learning rate of the keyframe's feed-forward network component to zero and for all other network components as one. Our model was built using the MatConvNet [79] library with the DagNN wrapper. Adaptive moment estimation (ADAM) [80] is utilized as the optimizer for our network parameters.

3.4 Experimental Analysis

In this section, we describe our experiments in detail and show the efficacy of our proposed method through a thorough analysis. All experiments are carried out on a desktop PC with Intel®Xeon(R) CPU E5-2690 v4 @ 2.60GHz, 16 Core with NVIDIA Quadro K2200 GPU. The proposed scheme is implemented using MATLAB R2020a. To test our model, we have used six test video sequences[1], namely, Akiyo, Coastguard, Foreman, Mother_daughter, Paris, and Silent. Results are compared with several state-of-the-art deep learning-based image CS methods ([29], [76], [30]) and video CS methods ([24], [75], [74], [4] and [20]).

3.4.1 Datasets and Performance Measures

For the deep recovery network, the training dataset is constructed using two standard HEVC video sequences, namely, Kimono and Cactus with resolution 1920×1080. In creating the training database, we employed 29 GOPs from the Kimono video and

25 GOPs from the Cactus video. These video sequences are divided into blocks of size 96×96. Consecutive 8 frames are taken to create patch GOPs of size $96 \times 96 \times 8$ by taking 8 patches having the same indices in their respective frames. All these patches were also augmented four times simultaneously. The aforementioned two videos were used to generate a dataset consisting of 1,10,000 GOPs for training. The validation dataset was similarly formed by taking 16 GOPs of two standard HEVC video sequences, namely, Ready and Basket. These two videos also have a resolution of 1920×1080. Similar to the training dataset, this validation dataset was also created by taking patches of size 96×96 from eight consecutive frames and augmenting them two times. We use PSNR and SSIM as quantitative measures for comparison following [29], [76], [30], [24], [75], [74] and [20].

3.4.2 Training and Parameter Selections

In our proposed network, we employ the pre-trained network of [20]. We have considered the parameters f, l, d, and n of our network in a way similar to [20]. According to [74], the size of the GOP is also set as 8. In our method, the frame-wise sampling is based on block compressive sensing, and the size of the block is set to $32 \times 32 \times 1$. During frame-wise initial recovery, the filter size is adjusted adaptively based on the sampling ratio. We used f=3, l=1, d=64, and n=4 in the deep restoration network for multilevel feature compensation. We investigate our proposed method for two sampling ratios 0.1 and 0.01. There are 100 epochs throughout the training. We consider different learning rates for different segments. The epochs 1-50, 51-80, and 81-100 have learning rates of 10^{-3}, 10^{-4}, and 10^{-5}, respectively.

3.4.3 Comparisons with SOTA Image and Video CS methods

We compare our method with both state-of-the-art (SOTA) deep learning-based image CS methods like ReconNet [29], ISTA-Net$^+$ [76], CSNet [30] and video CS methods like MC/ME [24], Video MH [75], RRS [74], PCS [4] and VCSNet [20].

In Table 3.1, we compare our proposed model with different video CS methods for the first two GOPs of six standard CIF video sequences concerning average PSNR and SSIM. For each method, the keyframes are sampled at 0.5 and all non-keyframes are sampled at different sampling ratios i.e 0.01, 0.05, and 0.1. We implement the PCS [4] method to sample each keyframe at SR=0.5 and each non-keyframe at

SR=0.01, 0.05, and 0.1, to ensure a fair comparison with our proposed model. With a SR of 0.01 for six videos, our double keyframe model outperforms MC/ME, Video MH, RRS, PCS and VCSNet-2 by an average of 7.75 dB, 6.07 dB, 11.26 dB, 5.35 dB and 0.45 dB respectively. Table 3.2 shows that the suggested double keyframe model outperforms MC/ME, Video MH, RRS, PCS, and VCSNet-2 at SR=0.05 by 3.67 dB, 3.17 dB, 5.73 dB, 6.58 dB, and 0.41 dB respectively. Our double keyframe model with SR 0.1 has an average gain of 2.98 dB over MC/ME, 2.55 dB over Video MH, 1.18 dB over RRS, 6.91 dB over PCS, and 0.45 dB over VCSNet-2 for a total of six videos. Similarly, our single keyframe model has a better average PSNR than MC/ME, Video MH, RRS, PCS and VCSNet-1. Our proposed single keyframe model has average PSNR improvement of 0.96 dB, 0.93 dB, and 0.92 dB in comparison to VCSNet-1 for SR 0.01, 0.05 and 0.1 respectively. For single keyframe-based CVS, we consider middle frame as keyframe, which compensates each non-keyframe equally. In VCSNet-1, first frame of the GOP is chosen as the keyframe and all non-keyframes are not compensated equally. So our single keyframe-based model has an improvement in average PSNR of approximately 1 dB over VCSNet-1 for all SRs. In addition, our single keyframe model has a better SSIM value than VCSNet-1.

Three deep learning-based image CS approaches, as well as two video CS i.e VCSNet-1 and VCSNet-2 methods, and our proposed method are compared in Table 3.2 for average PSNR and SSIM values of non-keyframes for the first two GOPs of six CIF video sequences. In all cases, the sampling ratio for keyframes is set at 0.5 and the sampling ratio for non-keyframes is set at 0.01 and 0.1. Foreman, Paris, and Coastguard videos contain more motion information, and our single keyframe-based model improves PSNR by 1.77 dB, 1.93 dB, and 1.22 dB, respectively, compared to VCSNet-1 for SR 0.01. When SR= 0.1, our suggested single keyframe model achieves a high PSNR for these recovered videos. Moreover, our suggested single keyframe model outperforms VCSNet-1 by an average of 1.09 dB and 1.03 dB for SR 0.01 and 0.1, respectively. Our proposed double keyframe approach shows gain by an average of 12.08 dB, 12.36 dB, 8.36 dB, 4.46 dB and 0.55 dB compared to ReconNet, ISTA-Net$^+$, CSNet, VCSNet-1 and VCSNet-2 respectively when the non-keyframes of six video sequences are sampled at a ratio of 0.01. Additionally, when compared to the non-keyframes of six video sequences sampled at a sampling ratio of 0.1, our proposed double keyframe approach outperforms ReconNet, ISTA-Net$^+$, CSNet, VCSNet-1 and VCSNet-2 by an average PSNR of 9.93 dB, 5.65 dB, 4.46 dB, 1.57 dB and 0.35 dB respectively. In comparison to VCSNet-2 at SR 0.01, our double

Table 3.1: Comparison of different video CS methods in terms of average PSNR and SSIM

Video	SR	MC/ME [24]		Video MH [75]		RRS [74]		PCS [4]		VCSNet-1 [20]		VCSNet-2 [20]		Our(Single Keyframe)		Our(Double Keyframe)	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Akiyo		27.15	0.8191	31.16	0.9161	25.09	0.7785	32.8	0.8087	33.67	0.9450	40.10	0.9834	34.41	0.9743	**40.54**	0.9852
Coastguard		21.65	0.4478	24.83	0.5536	22.25	0.4304	25.18	0.5809	26.64	0.6109	27.93	0.6775	27.60	0.6659	**28.35**	0.6777
Foreman		26.02	0.7504	27.84	0.8085	20.44	0.6471	25.24	0.6015	27.57	0.7846	28.89	0.8283	29.13	0.8233	**29.74**	0.8325
Mother_daughler	0.01	29.29	0.8174	32.59	0.8780	25.36	0.7142	34.99	0.8667	35.80	0.9301	39.80	0.9602	35.98	0.9367	**39.97**	0.9604
Paris		21.55	0.6616	20.64	0.6044	17.10	0.4143	21.56	0.5254	23.26	0.7593	26.55	0.8668	24.91	0.8191	**26.99**	0.8701
Silent		28.55	0.8327	27.22	0.7316	22.91	0.5908	28.82	0.7451	30.52	0.8708	34.71	0.9331	31.22	0.8668	**35.10**	0.9346
Average		25.70	0.7215	27.38	0.7487	22.19	0.5959	28.1	0.6881	29.58	0.8168	33.00	0.8749	30.54	0.8477	**33.45**	0.8767
Akiyo		36.59	0.9587	37.98	0.9635	33.95	0.9481	32.87	0.8093	38.03	0.9744	41.24	0.9829	39.15	0.9757	**41.51**	0.9831
Coastguard		27.07	0.6852	27.76	0.7205	25.17	0.5992	25.32	0.5837	27.94	0.6670	29.20	0.7324	28.99	0.7499	**29.77**	0.7327
Foreman		31.21	0.8590	32.36	0.8774	31.31	0.8977	26.14	0.6036	31.25	0.8682	32.91	0.8999	32.67	0.8945	**33.46**	0.9000
Mother_daughter	0.05	37.34	0.9365	37.67	0.9392	38.06	0.9409	35.26	0.8698	39.55	0.9583	40.72	0.9635	40.09	0.9585	**40.97**	0.9641
Paris		24.31	0.7409	24.27	0.7479	19.00	0.5508	22.07	0.5316	25.83	0.8528	27.78	0.8948	26.95	0.8731	**28.23**	0.8950
Silent		31.95	0.8702	31.44	0.8522	28.60	0.7874	29.34	0.7467	34.55	0.9286	36.18	0.9424	34.88	0.9292	**36.56**	0.9473
Average		31.41	0.8418	31.91	0.8501	29.35	0.7874	28.5	0.6908	32.86	0.8749	34.67	0.9026	33.79	0.8968	**35.08**	0.9037
Akiyo		38.77	0.9657	39.48	0.9693	41.45	0.9816	33.07	0.8109	39.72	0.9768	41.47	0.9815	41.23	0.9795	**41.97**	0.9838
Coastguard		28.09	0.7370	28.89	0.7814	29.35	0.7843	25.56	0.6386	29.38	0.7481	30.24	0.7876	30.15	0.7756	**30.59**	0.7898
Foreman		32.64	0.8803	33.94	0.8995	**35.50**	**0.9323**	26.56	0.6439	33.40	0.9087	34.28	0.9222	34.26	0.9198	34.74	0.9243
Mother_daughter	0.1	38.56	0.9449	38.93	0.9499	**42.02**	**0.9719**	35.77	0.8826	40.88	0.9645	41.43	0.9671	41.53	0.9675	41.65	0.9686
Paris		25.83	0.7778	25.58	0.7821	24.20	0.8213	22.91	0.5341	26.58	0.8779	28.23	0.9030	27.94	0.9055	**28.88**	0.9079
Silent		33.03	0.8841	32.69	0.8784	35.17	0.9162	29.46	0.7558	35.78	0.9372	36.44	0.9439	36.14	0.9398	**36.97**	0.9487
Average		32.82	0.8650	33.25	0.8767	34.62	0.9013	28.89	0.711	34.29	0.9022	35.35	0.9176	35.21	0.9146	**35.80**	0.9205
Average		29.98	0.8094	30.85	0.8252	28.72	0.7615	28.49	0.6966	32.24	0.8705	34.34	0.8984	33.18	0.8864	**34.78**	0.9003

keyframe model improves PSNR values by 1.27 dB and 0.63 dB for the Foreman and Paris videos, respectively. Also, the SSIM data demonstrate that our double keyframe model provides good reconstruction gain. The best values of Tables 3.1 and 3.2 are shown in bold. Fig.3.4 shows a frame-wise PSNR comparison for CSNet, VCSNet-1, VCSNet-2, and our suggested method on the first two GOPs of the video sequences Mother_daughter and Akiyo. Our suggested methods outperform CSNet and VCSNet. In Fig.3.5, we show the comparison of frame-wise PSNR for different deep learning-based image and video CS methods on 14 non-keyframes of first two GOPs of Akiyo video sequence for SR= 0.01 and 0.1. Our proposed double keyframe method shows better results than CSNet, ReconNet, ISTA-Net$^+$, VCSNet-1 and VCSNet-2. Our double keyframe method shows a better result than VCSNet-2 for all non-keyframes.

In Table 3.3, we compare the average time required for recovery of a GOP in a CIF video sampled at SR=0.1 for different video CS methods. Our proposed method takes less time as compared to MH-RTIK, MH-ST, MC/ME, Video MH, and PCS methods. Our proposed single and double keyframe method takes a little

Table 3.2: Comparison of different DL-based image CS methods and VCSNet in terms of average PSNR and SSIM of non-key frames

Video	SR	ReconNet [29]		ISTA-Net+ [76]		CSNet [30]		VCSNet-1 [20]		VCSNet-2 [20]		Our (Single Keyframe)		Our (Double Keyframe)	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Akiyo		22.54	0.6585	22.05	0.6789	26.80	0.8022	32.10	0.9384	39.45	**0.9823**	32.95	0.9368	**39.99**	0.9781
Coastguard		20.37	0.3291	19.87	0.3403	22.95	0.3890	24.79	0.5579	26.26	0.6340	26.01	0.6292	**26.54**	**0.6350**
Foreman		19.50	0.5422	19.32	0.5663	24.03	0.7091	25.81	0.7570	27.31	0.8069	27.58	0.8013	**28.58**	**0.8217**
Molher_daughter	0.01	22.73	0.6144	21.96	0.6284	27.93	0.7602	34.18	0.9212	38.75	**0.9555**	34.35	0.9169	**38.97**	0.9454
Paris		16.00	0.3148	16.62	0.3381	18.93	0.4647	22.04	0.7306	25.79	0.8535	23.97	0.7934	**26.42**	**0.8663**
Silent		21.38	0.4862	21.02	0.5013	24.17	0.6116	29.33	0.8568	34.13	0.9281	29.91	0.8525	**34.50**	**0.9331**
Average		20.42	0.4909	20.14	0.5089	24.14	0.6228	28.04	0.7937	31.95	0.8601	29.13	0.8217	**32.50**	**0.8633**
Akiyo		27.60	0.8041	33.16	0.9220	34.03	0.9451	39.01	0.9747	41.02	0.9801	40.74	0.9777	**41.70**	**0.9878**
Coastguard		23.53	0.4739	26.01	0.6087	27.63	0.6927	27.91	0.7147	28.90	0.7598	28.77	0.7515	**29.22**	**0.7955**
Foreman		25.63	0.7243	31.38	0.8769	31.30	0.8995	32.47	0.8989	33.47	0.9143	33.33	0.9115	**33.90**	**0.9580**
Mother_daughter	0.1	27.83	0.7571	34.08	0.8917	35.76	0.9233	39.98	0.9605	40.61	**0.9635**	40.69	0.9638	**40.99**	0.9617
Paris		19.40	0.4792	22.24	0.6877	23.64	0.7804	25.83	0.8662	27.72	0.8949	27.44	0.8945	**28.11**	**0.8999**
Silent		26.35	0.6737	29.16	0.7706	30.82	0.8359	35.34	0.9327	**36.09**	**0.9404**	35.76	0.9357	36.00	0.9365
Average		25.06	0.6521	29.34	0.7929	30.53	0.8462	33.42	0.8913	34.64	0.9088	34.45	0.9058	**34.99**	**0.9232**

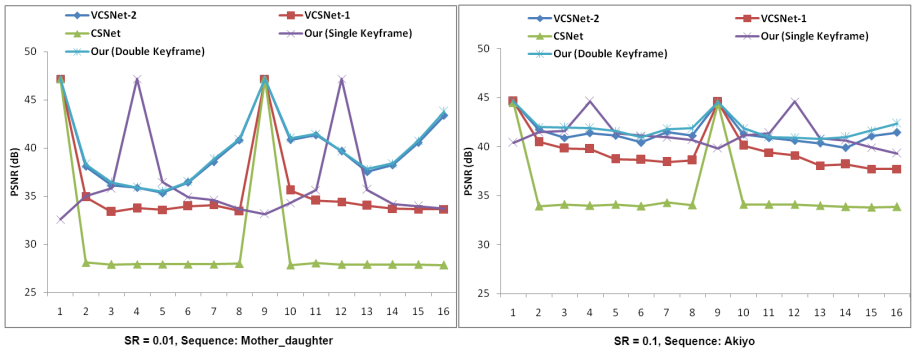


Figure 3.4: PSNR comparison of the first two GOPs from the videos Mother_daughter and Akiyo using CSNet, VCSNet, and our method.

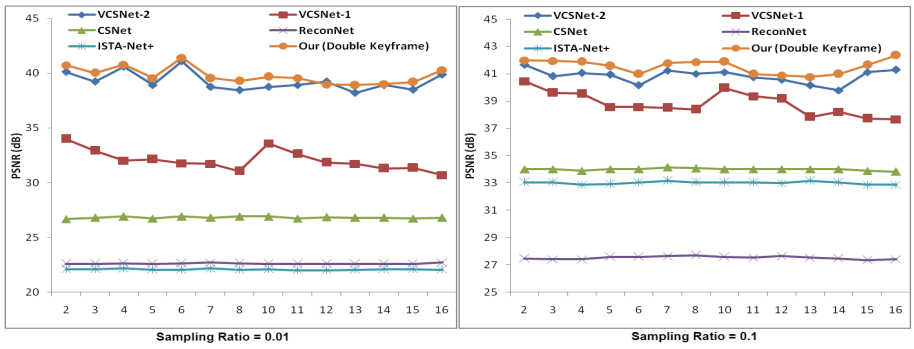


Figure 3.5: Comparison of PSNR of several deep learning-based CS methods on the first two GOPs of the Akiyo video's non-keyframes

bit more time for recovery than VCSNet-1 and VCSNet-2 respectively and this is because we consider both keyframe and nearer non-keyframes while compensating

Table 3.3: Comparison of average recovery time per GOP (in seconds) at SR=0.1

Method	Time
MC/ME [24]	237.578
MH-ST (m=40) [25]	9.775
MH-RTIK (m=40) [26]	17.075
MH-ST (m=200) [25]	41.575
MH-RTIK (m=200) [26]	88.8
Video MH [75]	24.8234
VCSNet-1 [20]	5.2418
VCSNet-2 [20]	8.1165
PCS [4]	15.1928
Our (Single Keyframe)	7.144
Our (Double Keyframe)	10.3072

m indicate numbers of hypotheses

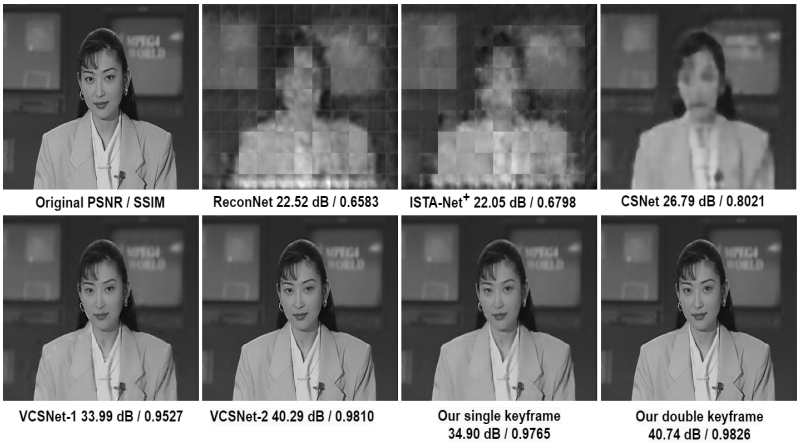


Figure 3.6: Video quality comparison of different image and video CS methods on the 2^{nd} frame of video sequence Akiyo for SR=0.01

non-keyframes. However, when compared to VCSNet1 and VCSNet2, our suggested approach outperforms them in terms of PSNR and SSIM of recovered frames. In Fig. 3.6, we show some qualitative visual quality comparison of different image and video CS methods on 2^{nd} frame of Akiyo video for sampling ratio = 0.01. Our proposed method shows better results as compared to other methods. In Fig. 3.7, we have shown the rate-distortion performance comparison [81] for different video compressive sensing methods. The PSNR and SSIM of recovered video frames are increasing with the increase in sampling ratio, and it is due to the decrease in distortion.

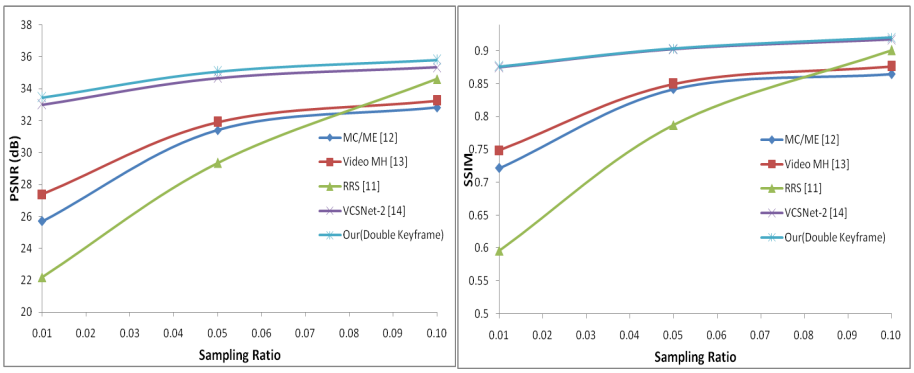


Figure 3.7: Rate-distortion performance for different video CS methods

3.4.4 Ablation Studies

We perform an ablation study to examine the individual contributions of multilevel feature compensation from only keyframes and both keyframes and non-keyframes on the quality of the recovered video. In Table 3.4, we show the average PSNR of the recovered video when the compensation is considered from only keyframes and both keyframes and neighboring non-keyframes. Our proposed method achieves better results in terms of average PSNR and SSIM when non-keyframes are compensated from both keyframes and neighboring non-keyframes than when only keyframes are considered during multilevel feature compensation.

Table 3.4: Average PSNR and SSIM values of two different compensation schemes with single keyframe and double keyframes for SR=0.01

First two GOPs of Video	Single Keyframe				Double Keyframe			
	Compensation from only keyframe		Compensation from both keyframe and non-keyframes		Compensation from only keyframes		Compensation from both key and non-keyframes	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Akiyo	33.71	0.9465	34.41	0.9743	40.11	0.9836	40.54	0.9852
Coastguard	26.89	0.6511	27.6	0.6659	27.95	0.6759	28.35	0.6777
Foreman	28.36	0.8046	29.13	0.8233	28.93	0.8284	29.74	0.8325
Mother_daughter	35.41	0.9167	35.98	0.9367	39.42	0.9548	39.97	0.9604
Paris	23.95	0.7817	24.91	0.8191	26.57	0.867	26.99	0.8701
Silent	30.56	0.8548	31.22	0.8668	34.59	0.9342	35.1	0.9346
Average	29.81	0.8259	30.54	0.8477	32.93	0.874	33.45	0.8767

3.5 Discussion

In this work, we present a video compression scheme with high-quality recovery. The proposed scheme applies deep compressive sensing-based video compression. The average quality of the recovered video frames improves with the inclusion of

multilevel feature compensation from the neighboring non-keyframes along with the keyframes. We compared our method with SOTA deep-learning-based image and video CS methods and found that our solution performs better in terms of PSNR and SSIM of the recovered frames.

Chapter 4

Joint Video Compression and Encryption

In this chapter, we propose a method for joint compression and encryption of video using parallel compressive sensing (PCS) and improved chaotic maps. The method is shown to be computationally efficient without compromising the quality of the recovered video. The processing of the video frames are carried out at group of pictures (GOP) level which consists of both reference and non-reference frames. We apply DCT for sparsification of the reference frames and permute the DCT coefficients to improve the quality of the reconstructed video. We introduce a Logistic Tent Infinite Collapse Map (LT-ICM) which has a higher chaotic range and complex behavior, unpredictability, and sensitivity towards initial values and controlling parameters. The measurement matrix for PCS is generated by using one-dimensional LT-ICM. A novel substitution method is introduced using circular shift, xor, and modular addition operations. We finally shuffle the substituted frames using frame dependent two-dimensional LT-ICM chaotic sequence. Experimental analysis, including comparisons with state-of-the-art approaches establishes the effectiveness of the proposed solution.

4.1 Introduction

Video encryption [82] [33] has evolved as an important area of research with applications in secured communication scenarios. In recent times, use of chaotic maps for video encryption has gained prominence [13] [58]. Chaotic maps are famous for generating pseudorandom sequences as they are highly unpredictable, sensitive to initial states and control parameters, and, are ergodic in nature. However, some of

the prevalent chaotic maps like Logistic, Tent and Sine have lower chaotic ranges. So, it is necessary to improve chaotic range for better security. Another typical problem for a video encryption method is that the size of the input video and that of the encrypted video are same. This leaves room for applying compression techniques, among which compressive sensing (CS) approaches have become popular [83]. Note that CS could be challenging though as it involves computation of Gaussian and Bernoulli measurement matrices, which can consume more memory and lead to high computational costs, especially for large data like a video [84]. Chaos based measurement matrix can be a better solution in such cases as it is generated from secret key and has faster recovery. In this work, we design a unified framework for efficient and secure video compression and encryption with parallel compressive sensing and improved chaotic maps. If the sampling process and the parallel CS reconstruction process are done column by column, the necessary storage and the computation required are much smaller than that of the conventional one. So, it makes the parallel CS very appropriate for processing videos with bulk data size.

Video encryption methods can be divided into three broad categories. The first category of methods encrypt video at frame level. Encrypted video frames exploit only spatial redundancy using techniques like chaotic map [13] [58] [55] [85], cellular automata [86] and quantum walk [87]. This class of methods has somewhat vulnerable security and yields low efficiency as input and encrypted frames have the same size. The second category of methods encrypt video frames considering both temporal and spatial redundancies at frame level [9]. These video encryption methods are more secure than the previous category. But, the size of the input and that of the encrypted video frames still remain the same leading to lower efficiency. The third group of encryption methods secure a video with lowest size. Compressive sensing based approaches are used for this kind of simultaneous compression and encryption of video data. Some of CS based cryptosystem methods can be found in [83] [35]. Encryption of video using only random measurement matrix is not perfectly secure. So, substitution and shuffling schemes are necessary after CS based sampling to make a video encryption scheme highly secure.

In this work, we propose a PCS and LT-ICM approach for joint compression and encryption of video at Group Of Pictures (GOP) level by exploiting both spatial and temporal redundancies. The first frame of each GOP is considered as a reference frame, and all other frames are considered to be non-reference frames. We take the difference between two consecutive frames in a GOP. Two dimensional Discrete Cosine Transform (2D DCT) is then applied to each reference frame. We then use

a new permutation on the DCT coefficients to distribute them evenly over all the columns. This permutation improves the quality of the reconstructed reference and the non-reference frames. We use separate measurement matrices for sampling reference and non-reference frames. The proposed measurement matrix (Φ), based on 1D LT-ICM, is secure, unpredictable, and sensitive to minor change in initial values or control parameters, or both. Then, we apply quantification to all of the sampled frames to bring each element's value within the range of 0 to 255. After that, we use a new substitution method to change the pixel values in each frame. Finally, each substituted frame is shuffled by a sort index-based permutation. We use a 2D LT-ICM which has a better chaotic range than the 1D LT-ICM for the substitution and permutation process. Our proposed PCS and chaotic maps-based video encryption are more secure due to an effective substitution method and a data-dependent permutation process. The chaotic maps produce chaotic sequences which in turn control PCS sampling, substitution, and permutation. So, the proposed encryption method can provide a high level of security for a video. The main contributions of the proposed video encryption algorithm are now highlighted below:

1. We present a unified scheme of compression and encryption of video at GOP level using PCS and LT-ICM. The proposed method is fast, secure and memory efficient.

2. We generate a measurement matrix (Φ) using LT-ICM. Besides the reduction in the size of video, the compressed video obtained using proposed LT-ICM based Φ matrix provides the first level of security. Also, the quality of the reconstructed video is found to be comparable to that obtained using other measurement matrices.

3. We further construct a 2D chaotic map using Logistic, Tent, and Infinite Collapse Map to improve the chaotic range and we employ it in our substitution method. The proposed substitution and the data-dependent chaotic sequence based shuffling operations ensure the second level of security in our scheme.

The rest of the work is organized as follows: In Section 4.2, we discuss the related works and point out their limitations. In Section 4.3, we describe our proposed joint video compression and encryption scheme in details. Section 4.4 presents the experimental results with analysis. Finally, the work is discussed in Section 4.5.

4.2 Related Work

This section begins with some representative video encryption methods. There are many works explored in [13] [58] [55] [85] [9] on video encryption using different chaotic maps. Chaotic maps used for video encryption in [13] [55] are not secure for a wide chaotic range. In [88], the authors used ICM chaotic map which has some discontinuity in chaotic ranges. To overcome this, Sethi et al. [9] proposed a new one-dimensional chaotic map (LT-ICM). But, there are some non-chaotic regions at lower values of the control parameter. In [85], a video encryption scheme is proposed by combining controlled XOR operations and an improved logistic map. The improved logistic map gives a better chaotic range than that of the logistic map. But, the chaotic range is between 0 to 4. An efficient technique, where, key-frames obtained from the video summarization process are encrypted in a probabilistic manner, is reported in [89]. A video data encryption technique using quantum walks-based substitution box is presented in [87]. In [58], authors introduced frame-level video encryption by using logistic sine coupling map (LSCM) keeping an inter-frame relation while generating the chaotic sequence. However, the methods reported in [58] [87] did not use temporal redundancy. In [90], authors proposed a dynamic key-based selective video encryption using a piece-wise linear chaotic map. However, their method is suitable for videos with less motion. The chaotic maps used for encryption [58] [85] [89] [90] have less chaotic range. A chaotic map with a high chaotic range is desirable for better security.

Since, our solution is based on joint video compression and encryption using compressive sensing we mention here some prominent works on CS-based cryptosystems. In [35], authors implemented a lightweight encryption method using CS and stream cipher along with encryption of MSB-bits. Bernoulli sensing matrix was generated utilizing stream cipher in [35] which has higher computational complexity. In [83], authors proposed a CS-based video encryption method using stream cipher. However, their method is suitable for a higher compression ratio (CR). There are a-few methods discussed in [68] [23] for video compression using CS theory. In [68] a scrambled block Hadamard matrix is used as the measurement matrix. But, reconstruction time is more for this method and reconstruction time at lower CR is larger than that of higher CR. Also, PSNR for reconstructed video frames is less at lower CR. A parallel CS-based video compression method for wireless multimedia sensor networks with a zigzag scan for permutation is presented in [23]. They have used a random Gaussian sensing matrix for the sampling process. The method used in [23]

takes less time to reconstruct video frames and has a higher PSNR of reconstructed video frames than the one utilized in [68]. In [91], the authors proposed a video compression, and encryption scheme without considering the temporal redundancy within the frames. They used 5D hyper-chaotic system and DNA encoding scheme in the encryption process. The blocking effect arises on the reconstructed frames due to the use of patch-based operation.

Now, we mention some of the salient works in CS-based video compression using multiple hypotheses (MH) prediction concept. Chen et al. [92] proposed a compressive sensing-based video compression method using reweighted Tikhonov regularisation for multiple hypothesis prediction reconstruction (MH-RTIK) that takes into account not only the similarity of hypotheses but also the effect of each hypothesis to improve resilience. They specifically used the effect of each hypothesis to synthetically reflect the influence of these parameters on MH prediction performance. They compared their result with different methods using Tikhonov-based algorithm (MH-TIK) [93], MH weighted Elastic net (MH-wEnet) [94] and MH-ST [95], which combines a Tikhonov regularization and a sparsity regularization for different videos under GOP=2. Each reference frame is sampled at a rate of 0.7, whereas non-reference frames are sampled at rates ranging from 0.1 to 0.5. However, MH-based methods take more time for reconstruction. The Gaussian random matrix is employed as the measurement matrix in all systems.

In this paragraph, we discuss several combined image compression and encryption methods based on compressive sensing. In [96], the authors proposed an approach where both compression and encryption processes are performed jointly within a single framework, thereby minimizing the complexity associated with conventional two-stage systems. The usage of a 4D hyper-chaotic system in their encryption technique increases the system's security. Ping et al. [97], proposed a unique visually secure image encryption approach that combines semi-tensor product compressed sensing and a partial block pairing-substitution algorithm. The experimental findings reveal that the system has excellent security and a large embedding capacity, which grows when the embedding is conducted on the spatial domain. However, the correlation between adjacent pixels of an encrypted image is higher for this method. Also, the entropy of encrypted images is less than of other existing methods.

Literature survey shows that there exist separate schemes for video encryption and video compression. We also observe that there are reported schemes for joint encryption and compression of images. However, to the best of our knowledge, there are very limited numbers of schemes available for video encryption and compression.

So, we propose an efficient and secure scheme for joint compression and encryption in this work.

4.3 Proposed Method

In this section, we discuss our proposed method in detail using three subsections. In the first subsection, we present the theory of parallel compressive sensing and show how it is being applied to our method. Permutation of DCT coefficients, generation of initial states and controlling parameters, generation of chaotic sequence, and measurement matrix are discussed next. The steps of our method are described in the last subsection with the help of three algorithms. The complete architecture of the proposed combined video compression and encryption scheme is shown in Fig. 4.1. Fig. 4.2 demonstrates the details of the encryption block (blue color block in Fig. 4.1) only where quantified compressed video frames are the inputs. The proposed encryption method consists of substitution followed by a sort index-based permutation. Fig. 4.3 shows the internal structure of the substitution block (yellow color block in Fig. 4.2).

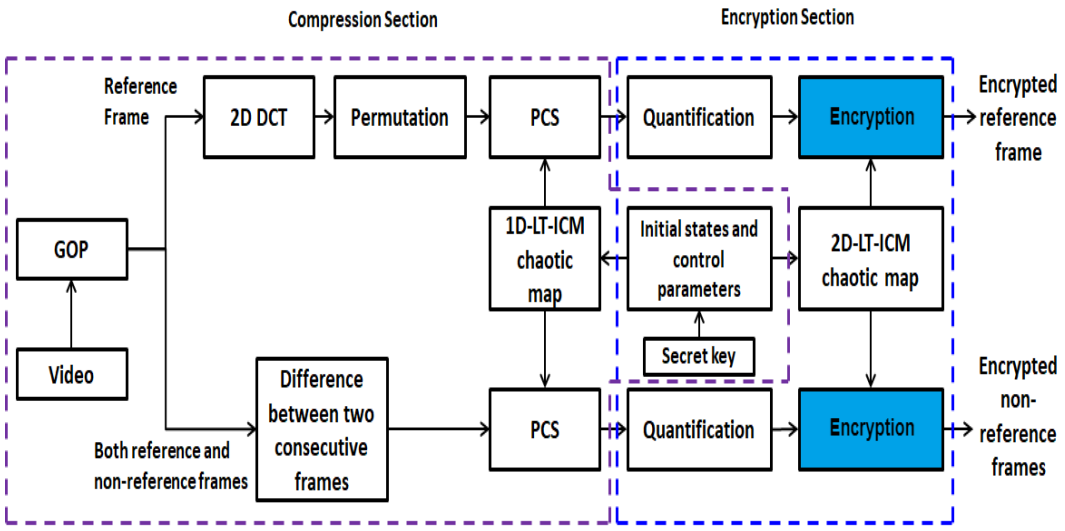


Figure 4.1: Block diagram of proposed joint video compression and encryption scheme

4.3.1 Parallel Compressive Sensing

In a CS method, the original signal can be reconstructed from very less number of sample data or sparse data. The recovery is possible as sparse data can be

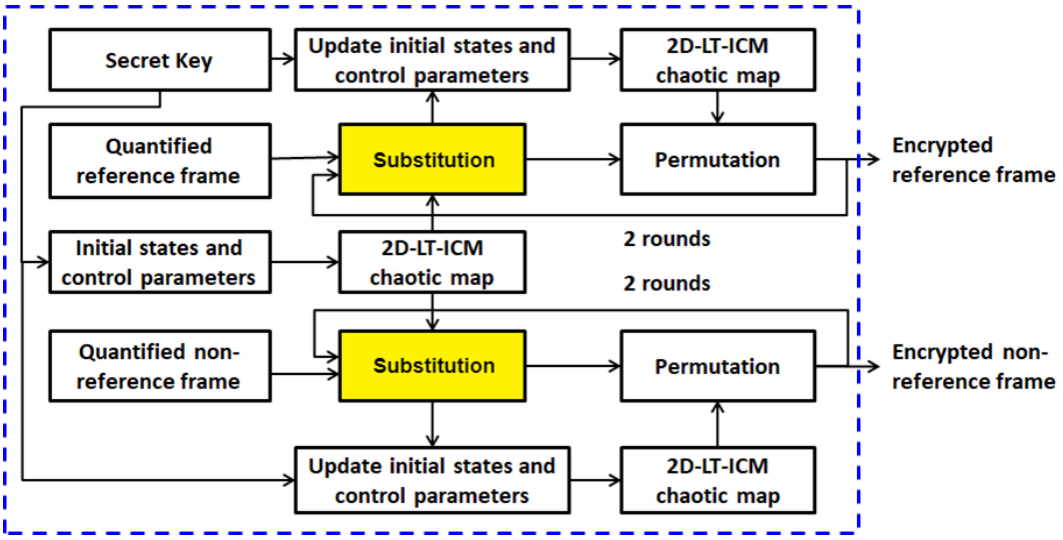


Figure 4.2: Block diagram of encryption scheme

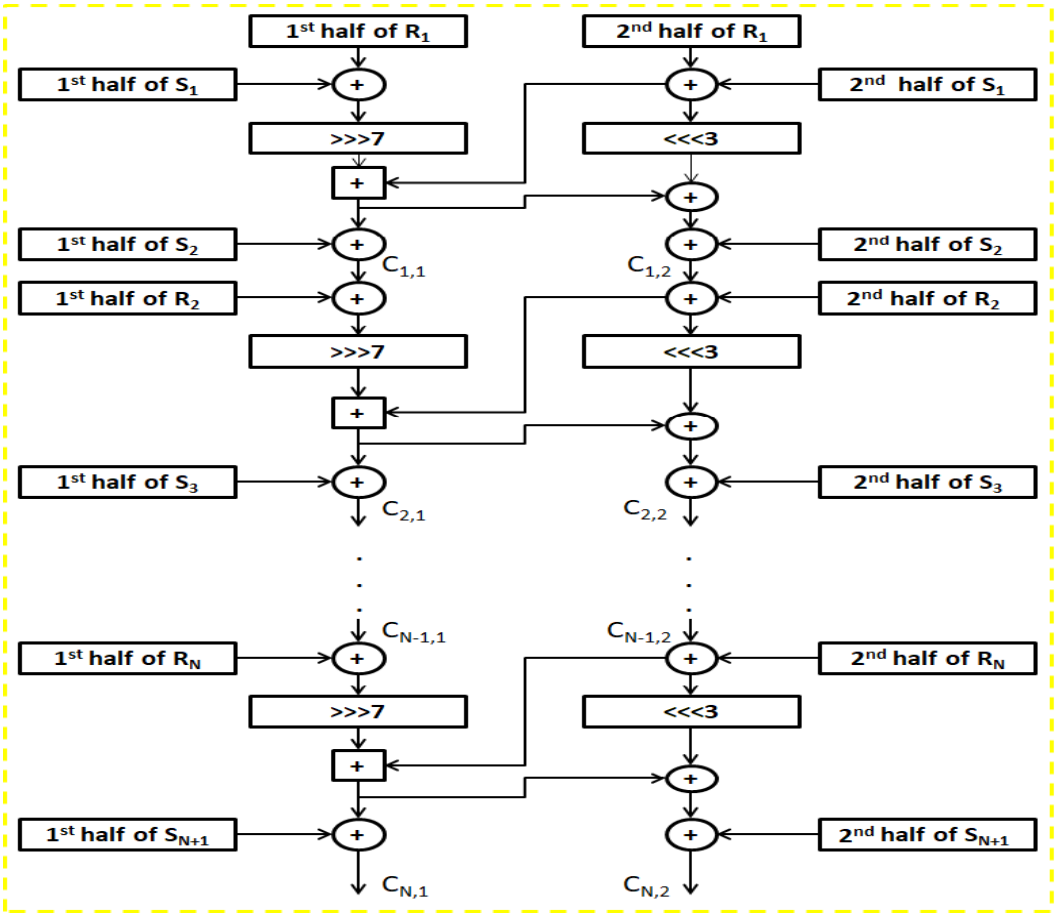


Figure 4.3: Block diagram for proposed substitution method

expressed into a small number of projections. Generally, a higher dimensional signal is converted into a 2D signal which is then processed column-wise to achieve parallel

compressive sensing(PCS). We apply PCS column-wise for each frame in the GOP. Let the sparse signal of x_i be s_i of size $N \times 1$ and the measurement matrix be Φ of size $M \times N$, where, $M << N$. Then, sampling of the signal x_i can be expressed as:

$$y_i = \Phi \times x_i = \Phi \times \Psi \times s_i = \Theta \times s_i \tag{4.1}$$

where Ψ is the orthonormal basis matrix of size $N \times N$, Θ is the sensing matrix of the same size as that of Ψ, y_i is the measurement vector of size $M \times 1$ and $i = 1, 2, 3, \cdots, N_c$ (number of columns) in a frame. The expression for compression ratio (CR) is M/N. To get a sparse signal we have to convert the signal into its frequency domain. We use DCT basis matrix to get the sparse representation of the original signal.

The recovery of *K-sparse* signal (a sparse signal having K non-zero elements) from M number of measurements is possible if $M > K \log_2(N/K)$ [98] and the Φ matrix satisfies the restricted isometry property (RIP) [99]. Further, Φ matrix has to be incoherent with Ψ matrix to ensure faithful reconstruction. The sparse signal can be recovered by the following manner:

$$\min \|s_i\|_0, \text{ s.t. } y_i = \Theta \times s_i \tag{4.2}$$

where $\|s_i\|_0$ denotes l_0 norm of s_i. Obtaining l_0 norm becomes a NP-hard problem. So, to overcome this a convex optimization method is considered which uses l_1 norm for recovery of sparse data. The reconstruction can be performed using the basis pursuit algorithm, which addresses an l_1-norm minimization problem. The algorithm may be applied in a column-wise fashion to recover sparse data from the measurement data. PCS shows a better computational complexity [23] and it takes controlling parameters and initial state as secret key for forming the measurement matrix.

Since most of the energy in a piecewise smooth video frame is in the low frequencies, the 2D DCT coefficients of the video frame often have most of the large coefficients in the top left corner and most of the small coefficients in the bottom right corner. During the recovery process of parallel compressive sensing (PCS), the video data is reconstructed column by column from the measurements. If the sparse DCT coefficient data is not permutated then most non-zero significant coefficients are clustered around the top left corner of the transformed matrix. The first few columns in the transformed frame consist of more non-zero significant coefficients and the rest of the columns of the transformed frame have coefficients with very

less or zero value. So there is a non-uniform distribution among different columns resulting decrement in reconstructed average PSNR. After the permutation of DCT coefficients, the energy is distributed more uniformly across columns, which may be thought of as a rough interpretation of the sparsity vector if all non-zero entries in the 2D signal have the magnitude of the same order. The authors in [23] suggested that when the DCT coefficients are permutated, the average PSNR of recovered video is higher than when they are not permutated. This is because after proper permutation each column looks equally sparse and the columns are uniformly distributed. In Fig. 4.4, we have shown the DCT coefficients of a frame for the Foreman video before and after the permutation. The left side image in Fig. 4.4 has non-uniform distribution in most of the columns whereas the right side image has uniform distribution for every column. In our proposed scheme, we apply DCT to just reference frames in each group of pictures (GOP) since they are not sparse, and then apply permutation to DCT coefficients. The difference between two successive frames is sparse enough. So, we do not apply DCT and permutation to non-reference frames. The recovered reference frame has better quality since permuted DCT coefficients are sampled at a higher sampling rate. The recovered non-reference frame depends on the already recovered reference frame. As a result, we apply permutation to the DCT coefficients, which not only improves the quality of the reference frame but also the non-reference frames.

As an example, we show that our permutation is useful for 2D DCT coefficients. We Consider a 2D signal $X \in R^{M \times N}$ and apply 2D DCT on it. Let M=4 and N=4 for the given matrix X. The coefficients of DCT matrix is arranged in descending order from the first row to the last row with respect to its absolute value such that $|p_1| > |p_2| > |p_3| > |p_4| \ldots\ldots |p_{15}| > |p_{16}|$. By doing this all columns are sparse and every column is evenly distributed. With this permutation, the average PSNR of recovered videos is enhanced from 3 dB to 9 dB than that of recovered videos without permutation for different compression ratios and this is possible because all the columns are sparse enough to give better results. We show how to form 2D DCT coefficient matrix after permutation in Eq.(4.3).

$$P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \\ p_{13} & p_{14} & p_{15} & p_{16} \end{bmatrix} \tag{4.3}$$

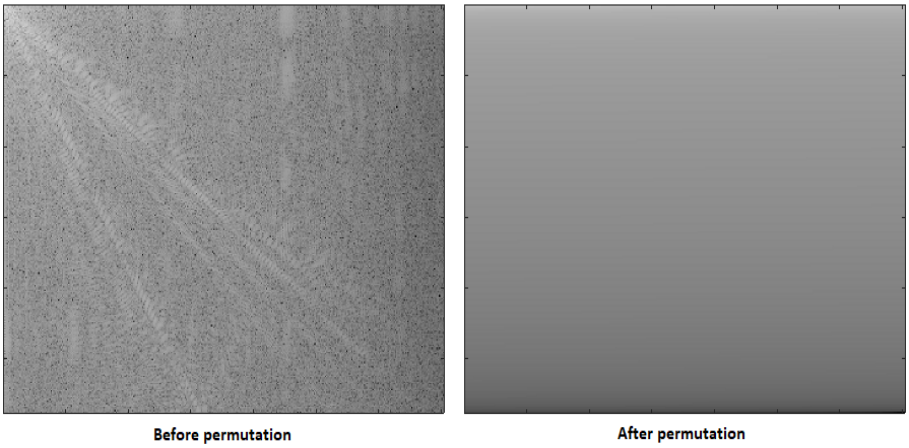


Figure 4.4: Distribution of DCT coefficients before and after permutation

4.3.2 Secret Key

The length of secret key K is 512 and it is utilized to produce the initial states and controlling parameters for both the 1D LT-ICM and 2D LT-ICM chaotic systems. The secret key consists of 16 parts and each part takes 32 bits. We generate $(x_R, a_R, x_{NR}, a_{NR})$ for 1D LT-ICM system used in compression, $(x_R^1, y_R^1, a_R^1, x_{NR}^1, y_{NR}^1, a_{NR}^1)$ for 2D LT-ICM (first round encryption) and $(x_R^2, y_R^2, a_R^2, x_{NR}^2, y_{NR}^2, a_{NR}^2)$ for 2D LT-ICM (second round encryption) from shared secret key. The initial states of the reference frame and non-reference frames are represented by (x_R, y_R) and (x_{NR}, y_{NR}) respectively whereas a_R and a_{NR} are the corresponding controlling parameters. The generation of initial states and controlling parameters is shown in Algorithm-5.

4.3.3 Chaotic Sequence Generation

The logistic map, tent map and iterative chaotic map with infinite collapse are commonly used 1D chaotic maps. Mathematically, they are represented as:
Logistic map:

$$x_{i+1} = r_1 x_i (1 - x_i) \tag{4.4}$$

Tent map:

$$x_{i+1} = \begin{cases} 2r_1 x_i & \text{if } x_i < 0.5 \\ 2r_1(1 - x_i) & \text{if } x_i \geq 0.5 \end{cases} \tag{4.5}$$

Algorithm 5 Generation of initial states and controlling parameters

Input: (Secret key B of length 512 bits)
Output: Initial states and controlling parameters

1: $x_R = \sum_{i=1}^{32} B_i \times 2^{-i}$
2: $x_{NR} = \sum_{i=1}^{32} B_{i+32} \times 2^{-i}$
3: $a_R = \sum_{i=1}^{32} B_{i+64} \times 2^{-i} + 25$
4: $a_{NR} = \sum_{i=1}^{32} B_{i+96} \times 2^{-i} + 33$
5: $x_R^1 = \sum_{i=1}^{32} B_{i+128} \times 2^{-i}$
6: $y_R^1 = \sum_{i=1}^{32} B_{i+160} \times 2^{-i}$
7: $a_R^1 = \sum_{i=1}^{32} B_{i+192} \times 2^{-i} + 20$
8: $x_{NR}^1 = \sum_{i=1}^{32} B_{i+224} \times 2^{-i}$
9: $y_{NR}^1 = \sum_{i=1}^{32} B_{i+256} \times 2^{-i}$
10: $a_{NR}^1 = \sum_{i=1}^{32} B_{i+288} \times 2^{-i} + 22$
11: $x_R^2 = \sum_{i=1}^{32} B_{i+320} \times 2^{-i}$
12: $y_R^2 = \sum_{i=1}^{32} B_{i+352} \times 2^{-i}$
13: $a_R^2 = \sum_{i=1}^{32} B_{i+384} \times 2^{-i} + 30$
14: $x_{NR}^2 = \sum_{i=1}^{32} B_{i+416} \times 2^{-i}$
15: $y_{NR}^2 = \sum_{i=1}^{32} B_{i+448} \times 2^{-i}$
16: $a_{NR}^2 = \sum_{i=1}^{32} B_{i+480} \times 2^{-i} + 32$

ICM map:

$$x_{i+1} = \sin(r_1/x_i) \tag{4.6}$$

where r_1 is the controlling parameter and $r_1 \in [0, 1]$ for Logistic, Tent map and $r_1 > 0$ for ICM map. An iterative chaotic map with infinite collapse is presented in [88]. Sethi et al. [9] proposed a one dimensional LT-ICM to overcome the limitation in [88]. Mathematically, a 1D LT-ICM is expressed as:

$$x_{i+1} = \begin{cases} \sin(r_1/(2x_i r_1 + (1 - r_1)x_i(1 - x_i) + 0.5)) & \text{if } x_i < 0.5 \\ \sin(r_1/(2(1 - x_i)r_1 + (1 - r_1)x_i(1 - x_i) + 0.5)) & \text{if } x_i \geq 0.5 \end{cases} \tag{4.7}$$

We propose a chaotic 2D LT-ICM for further improvement of chaotic range. The proposed map has higher entropy throughout the chaotic range of control parameter. The chaotic map is designed to address the limitations of existing chaotic maps exhibiting weak chaotic and low dynamical behaviors. The phase space diagram and bifurcation diagram for 2D LT-ICM are shown in Fig. 4.5. The proposed map has better chaotic behaviour at lower range of control parameter than that of 1D LT-ICM [9].

Mathematically, a 2D-LT-ICM can be represented as:

$$\begin{cases} x_{i+1} = \begin{cases} \sin(r_1/(0.1y_i r_1 + (1-r_1)x_i(1-x_i) + 0.1)) & \text{if } x_i < 0.5 \\ \sin(r_1/(0.1(1-y_i) r_1 + (1-r_1)x_i(1-x_i) + 0.1)) & \text{if } x_i \geq 0.5 \end{cases} \\ y_{i+1} = \begin{cases} \sin(r_1/(0.1x_{i+1} r_1 + (1-r_1)y_i(1-y_i) + 0.1)) & \text{if } x_i < 0.5 \\ \sin(r_1/(0.1(1-x_{i+1}) r_1 + (1-r_1)y_i(1-y_i) + 0.1)) & \text{if } x_i \geq 0.5 \end{cases} \end{cases} \tag{4.8}$$

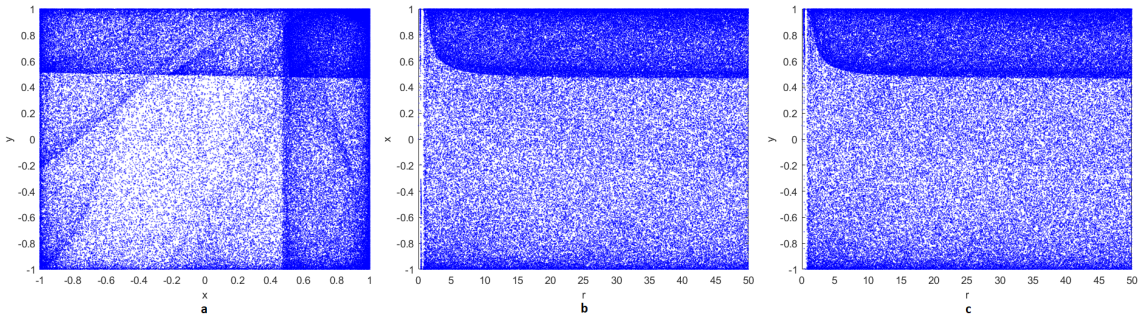


Figure 4.5: 2D LT-ICM: (a) Phasor diagram (b,c) Bifurcation diagrams

Table 4.1: Quantitative comparison between 1D LT-ICM and 2D LT-ICM chaotic maps

Metrics	**1D LT-ICM**	**2D LT-ICM**
Dimension	1	2
Phase space	1D	2D (Larger state space)
Maximum Lyapunov exponent	3.23	9.72
Sample entropy	1.40	2.01

Also, a quantitative comparison between the 1D LT-ICM and 2D LT-ICM chaotic maps is presented in Table 4.1. The comparison shows that the 2D LT-ICM exhibits improved chaotic performance, as reflected by its higher maximum Lyapunov exponent and sample entropy compared with the 1D LT-ICM.

4.3.3.1 Sensitivity Analysis of Chaotic Sequence

Sensitivity to initial state is an important property of the chaotic map. At first level, a video is secured using CS based approach where of measurement matrix is generated from 1D LT-ICM. The 1D LT-ICM is very sensitive to initial state and controlling parameter. We generate three chaotic sequences (seq 1, seq 2 and seq 3) of length 50000 by 1D LT-ICM considering minor change in control parameter and initial state. The trajectories of first 50 iterations are shown in Fig. 4.6 for the

proposed 1D LT-ICM when the change in control parameter and initial state is very small. We observe that even for the minor change in control parameter and initial state the trajectories look quite different from each-other.

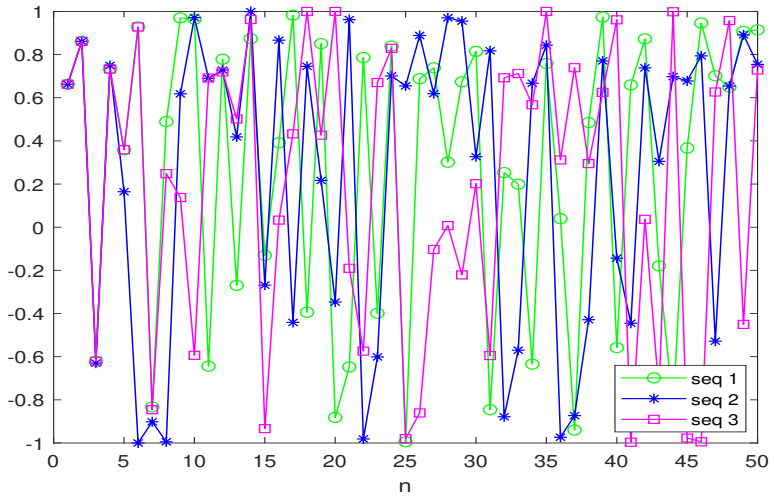


Figure 4.6: Three trajectories seq 1, seq 2, seq 3 with initial value and control parameter as (0.3,30), (0.30001,30) and (0.3,30.00001) respectively

4.3.3.2 Measurement Matrix generation

We generate measurement matrix using 1D LT-ICM chaotic system presented in [9]. A chaotic sequence S is generated using Eq.(4.7) of length $MNd + n_0$ where M, N, d, n_0 indicates number of rows, columns, sampling distance and a constant respectively. Here, we sample S with a sampling distance $d = 5$ by $Z_i = S_{3000+i*d}$, where, $i = 1, 2, \cdots, M \times N$. Measurement matrix is generated using Z in the following manner:

$$\Phi = \sqrt{2/M} \begin{bmatrix} Z_1 & Z_{M+1} & ,\dots & Z_{(N-1)M+1} \\ Z_2 & Z_{M+2} & \dots & Z_{(N-1)M+2} \\ \vdots & \vdots & \ddots & \vdots \\ Z_M & Z_{2M} & \dots & Z_{MN} \end{bmatrix} \tag{4.9}$$

where $\sqrt{2/M}$ is the normalization factor. The measurement matrix generated using 1D LT-ICM can recover the original signal when it satisfies the restricted isometry property and is incoherent with sparse basis matrix Ψ. The elements chosen from the chaotic sequences should be statistically independent while forming the measurement matrix. We ensure this using higher order correlation [98] with a sample distance

of 5. We divide the interval between -1 to 1 into a number of fixed intervals and calculate the number of samples within each interval for both the original chaotic sequence and its sampled version. Joint probability density using both sequences is obtained next. For the logistic map and the tent map with sampling distance $d = 5$, the joint probability density looks more like a non-uniform distribution which tells that the samples x_n and x_{n+d} can be highly correlated. But for the LT-ICM, the joint probability density looks more uniform which indicate that x_n and x_{n+d} are uncorrelated. The simulation result of joint probability density of x_n and x_{n+d} for Logistic, Tent and LT-ICM are shown in Fig. 4.7. In addition, we have shown a scatter diagram for 1D LT-ICM in Fig. 4.8.

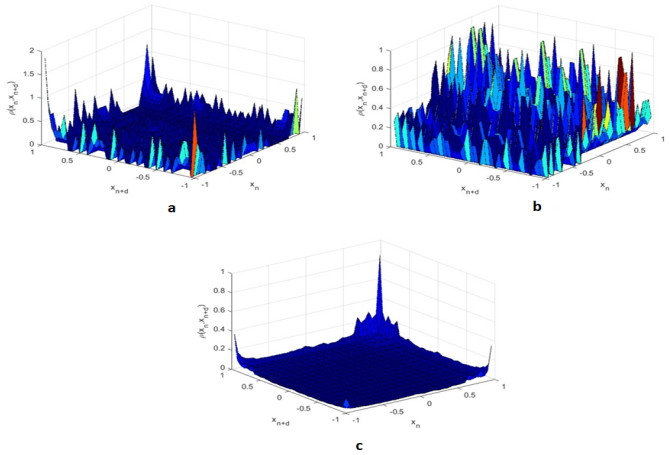


Figure 4.7: Probability density function of a) Logistic b) Tent and c) 1D LT-ICM

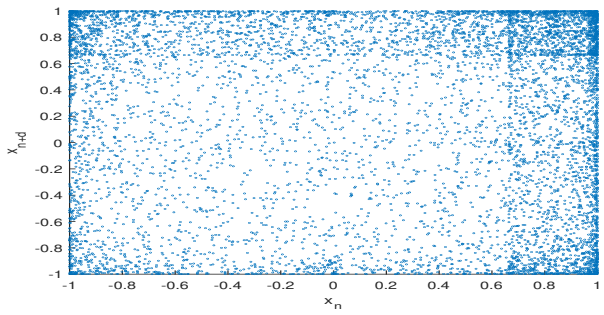


Figure 4.8: Scatter diagram of 1D LT-ICM

4.3.3.3 A New Substitution method

Substitution method is used to change the pixel values with the help of some pseudorandom sequence so that attacker can not know the exact values. We use modular addition, circular shift and xor in the substitution method which makes our scheme more secure. The columns of the quantified frame are divided into two parts. A chaotic random matrix is generated of the same size plus one more row than that of the quantified frame. The columns of the generated chaotic matrix are also divided into two parts.

We apply substitution in the first row in a frame which is described in Eq.4.10.

$$\begin{cases} C_{i,1} = S_{i+1,1} \oplus (((R_{i,1} \oplus S_{i,1}) >>> 7) \boxplus (R_{i,2} \oplus S_{i,2})) \\ C_{i,2} = S_{i+1,2} \oplus (((R_{i,1} \oplus S_{i,1}) >>> 7) \boxplus (R_{i,2} \oplus S_{i,2})) \oplus ((R_{i,2} \oplus S_{i,2}) <<< 3) \end{cases} \tag{4.10}$$

For all other rows, the substitution process is described in Eq.4.11.

$$\begin{cases} C_{i,1} = S_{i+1,1} \oplus (((R_{i,1} \oplus C_{i-1,1}) >>> 7) \boxplus (R_{i,2} \oplus C_{i-1,2})) \\ C_{i,2} = S_{i+1,2} \oplus ((R_{i,1} \oplus C_{i-1,1}) >>> 7) \boxplus (R_{i,2} \oplus C_{i-1,2}) \oplus ((R_{i,2} \oplus C_{i-1,2}) <<< 3) \end{cases} \tag{4.11}$$

R, S, and C indicate row of the sampled frame, chaotic matrix, and substituted matrix, respectively. The circular right shift and circular left shift are indicated by $>>>$ and $<<<$ respectively. The xor and modular addition are represented by $\oplus$ and $\boxplus$ respectively.

4.3.4 Steps of the Proposed Method

In this section, we describe the steps of our proposed method. We first divide the videos into a fixed-length group of pictures (GOPs) which consists of both reference and non-reference frames. We consider the difference between two consecutive frames for non-reference frames. We consider GOPs of size 2 and 4 frames for our experiment. Each reference frame is represented in sparse form by applying 2D DCT. No further sparsification is applied in differenced non-reference frames as they are sparse. The DCT coefficients for each reference frame in the GOP are permuted in descending order with respect to their absolute values such that all coefficients are evenly distributed and each column become sparse. Next, a measurement matrix

Φ is generated using Eq.4.9. The Φ and Ψ matrices are used to form the sensing matrix Θ. We sample all columns in both frames using PCS considering Eq.4.1. This leads to measurement data with a reduced size for each column. We consider a separate measurement matrix for reference and non-reference frames. Measurement data for all frames are quantified so that each element will take the values within the range 0 to 255. We used Eq.4.13 for quantification of sample data and inverse quantification is done by using Eq.4.14. A substitution method is employed next to change the pixel values randomly for each frame in a GOP using modular addition, circular shift and xor operations. The use of nonlinear operation makes our substitution method more secure. We employ 2D LT-ICM chaotic sequence in the substitution step. After the substitution step, the resultant frame is reshaped into 1D array. The initial states and controlling parameters are updated by using the pixel values received from the substitution result. A chaotic sequence of the same length of reshaped substitution frame is generated using 2D LT-ICM. The substituted result is permuted by the indices of generated chaotic sequence. Similarly, substitution and permutation are used to encrypt non-reference frames. The initial states and controlling parameters are different than that of the reference frame. The initial states and controlling parameters are updated before giving to 2D LT-ICM chaotic sequence generator as follows:

$$\begin{aligned} ss &= \sum_{1}^{MN} 10^{-\log_2(D(i)+255)} \\ s_0 &= ss/(\text{number of elements in D}) \\ x &= x + s_0 \\ y &= y + s_0 \\ a &= a + s_0 \end{aligned} \tag{4.12}$$

where D is the substituted frame in 1D form, and s_0 is the parameter used for updating x, y, and a.

$$P_Q = round(((P - P_{min}) \times 255)/(P_{max} - P_{min})) \tag{4.13}$$

$$P = (((P_{max} - P_{min}) \times P_Q)/255) + P_{min} \tag{4.14}$$

where P_Q is the quantified version of P and P_{min} and P_{max} are the minimum and maximum values of P.

To recover original video frames, first decryption is performed, followed by inverse quantification and decompression using the recovery technique. In decryption, inverse permutation is performed first, followed by inverse substitution. Then we apply the inverse of quantification process to bring back the original DCT coefficients for the reference frame. We retrieve the sparse data for both reference and non-reference frames during decompression using the l_1-magic package [100]. Then, for the reference frame, we do the inverse operation of the permutation so that the DCT coefficients will take their original position. Then 2D IDCT is applied to sparse data to get reference frame. Similarly, after recovering sparse data for non-reference frames, addition is performed between two consecutive frames to recover non-reference frames

Algorithm 6 A Joint Video Compression and Encryption Algorithm

Input: (Original video, Secret key, CR), where CR is the compression ratio
Output: Compressed and Encrypted video

1: Segment the video into GOPs of fixed length
2: Divide a GOP into reference and non-reference frames
3: Choose measurements ratio between reference frame to non-reference frame 4:1 for 2 frames in a GOP and 3:1 for 4 frames in a GOP respectively
4: Compress each GOP using Algorithm-7
5: Encrypt each compressed GOP employing Algorithm-8
6: Apply steps 4 to 5 for for all remaining GOPs in the video
7: Combine all compressed and encrypted GOPs

Algorithm 7 Video Compression Algorithm at GOP Level

Input: (Original GOP level video, $N, CR, x_R, a_R, x_{NR}, a_{NR}$), where N is the length of GOP and CR is the compression ratio
Output: Compressed GOP

1: Compute difference between two consecutive frames considering first frame of each GOP as reference frame
2: Compute 2D DCT of reference frame
3: Permute the DCT coefficients ▷ (Eq.4.3)
4: Generate Φ_1 matrix using 1D LT-ICM and x_R, a_R ▷ (eqs. 4.7, 4.9)
5: Perform PCS on each column of permuted reference frame using Φ_1 matrix ▷ (eq 4.1)
6: Generate Φ_2 matrix using 1D LT-ICM and x_{NR}, a_{NR} ▷ (eqs. 4.7, 4.9)
7: Perform PCS on each column of non-reference difference frames using Φ_2 matrix ▷ eq 4.1)

Algorithm 8 Video Encryption Algorithm at GOP Level

Input: (Compressed GOP, $x_R^1, y_R^1, a_R^1, x_{NR}^1, y_{NR}^1, a_{NR}^1, x_R^2, y_R^2, a_R^2, x_{NR}^2, y_{NR}^2, a_{NR}^2$)
Output: Encrypted GOP

1: Quantify compressed samples within the range between 0 to 255 ▷ (Eq. 4.13)
2: Generate first chaotic matrix of size same as of quantified reference frame considering (x_R^1, y_R^1, a_R^1) as secret key
3: Perform substitution of reference frame using generated first matrix. ▷ Eqs. 4.10, 4.11
4: : Arrange all the substituted samples of reference frame in an array
5: Update (x_R^1, y_R^1, a_R^1) using the substituted reference frame ▷ (Eq. 4.12)
6: Generate second chaotic sequence of length same as of substituted reference frame considering updated (x_R^1, y_R^1, a_R^1)
7: Shuffle the substituted reference frame using 2D LT-ICM based second chaotic sequence ▷ (Eq. 4.8)
8: Reshape shuffled result in 2D matrix
9: Apply steps 2 to 8 for second round encryption using (x_R^2, y_R^2, a_R^2) as secret key
10: Apply steps 2 to 9 for non-reference frames considering initial state and controlled parameters as $(x_{NR}^1, y_{NR}^1, a_{NR}^1, x_{NR}^2, y_{NR}^2, a_{NR}^2)$

4.4 Experimental Results

In this section, we analyze the performance of our method. Different YUV video (Quarter Common Intermediate Format) sequences of 100 frames were used to evaluate our technique, including Akiyo, Foreman, Coastguard, Silent, Bus, Suzie, Salesman and News. We consider Foreman, Coastguard, Hall, and Container video (Common Intermediate Format) of size 352×288 in our experiment. We also employ various test videos mentioned in [91]. All experiments are performed on a desktop PC with an i7-9700H CPU having 3.0 GHz clock speed and 8 GB RAM. We used MATLAB R2019b in the Windows 10 environment for our implementation. In the case of CIF videos and videos listed in [91], we consider CR=0.7 for reference frames and CR=0.1 to 0.5 for non-reference frames.

4.4.1 Histogram Analysis

The histogram is used to visualize the distribution of pixel intensity within a video frame. There must be an uniform distribution in histogram of encrypted video frame for better security. Histogram of selected encrypted frames for different videos are shown in Fig. 4.9. The histogram of encrypted frames look more or less uniform. So, our proposed method can defend different statistical attacks.

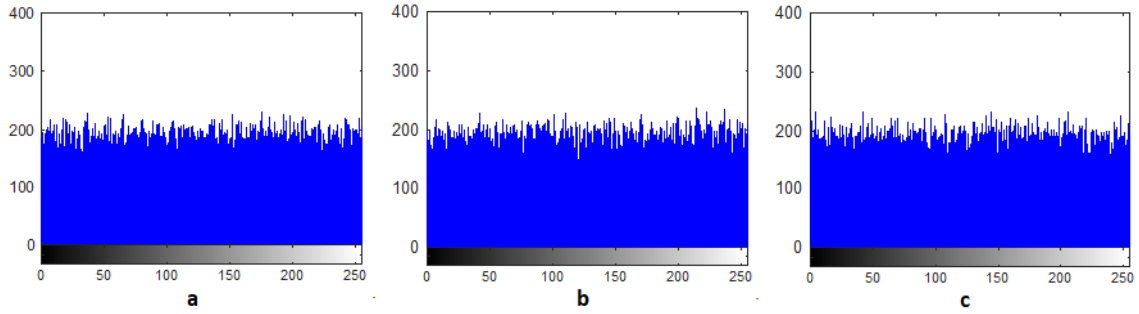


Figure 4.9: Histograms of encrypted frame for video (a) Foreman (b) Coastguard and (c) Container

4.4.2 Key Space and Key Sensitivity Analysis

In the proposed scheme, we consider the secret key length is 512. This makes the total key space as 2^{512} which is much greater than 2^{128}, the minimum key space requirement for any secure encryption scheme. So, our method can defend exhaustive brute force attack. Table 4.2 shows comparison of key space with some existing methods. Decryption is not possible even with a very minor change in the secret key, which makes the proposed system highly sensitive to key variations and enhances its security. A minor change in secret key will produce a different kind of noisy reconstructed frame than that of original frame which is shown in Fig. 4.10. In the case of a known-plaintext attack, the attacker knows both the plaintext and the ciphertext. If some part of the input frames is known in a chosen-plaintext attack, the attacker can get encrypted frames. The use of nonlinear operations in substitution makes our scheme robust agnaist chosen-plaintext attack. Any small change in the secret key makes decryption impossible as the chaotic sequence generation depends on substituted data for updating the initial states and controlling parameters for 2D LT-ICM chaotic system. As a result, our method can defend against both known and chosen-plaintext attacks.

Table 4.2: Key space of different schemes

Methods	Key space
Ranjithkumar et al. [55]	2^{212}
El-Latif et al. [87]	2^{305}
Sethi et al. [9]	2^{186}
Muhammad et al. [89]	2^{299}
Khlif et al. [90]	2^{147}
Ping et al. [97]	2^{512}
Our method	2^{512}

Figure 4.10: Key sensitivity test with Akiyo video (a) Original frame (b) Compressed and encrypted frame (c) recovered frame using exact key (d) recovered frame using a key that differs by one bit from the exact key

4.4.3 Comparison with State-of-the-Art Methods

Finally, we compare our result with some state-of-the-art video compression [68] [23] [92] [93] [94] [95] and encryption [58] [83] [55] [85] [87] [89] [96] [91] [101] methods.

4.4.3.1 Performance Analysis of Video Compression

The typical YUV sequence Akiyo, Foreman and Coastguard in QCIF format was utilized in the simulation. The sparsifying basis in our strategy is the DCT basis. The sparse video data is recovered using basis pursuit method available in l_1-magic package [100]. We compare the reconstruction time and mean PSNR of reconstructed video with some existing methods to demonstrate the benefit of the video compression scheme presented in Section 3. The results are shown in Table 4.3. Our proposed method gives better PSNR and for all CR values, the reconstruction time is less than the method used in [68] [23]. The permutation of DCT coefficients before sampling increases the average PSNR of reconstructed video than that of without permutation. In Fig. 4.11, the average recovered PSNR is shown concerning different CR with and without permutation for different videos. The permutation of DCT coefficients by arranging the absolute values in descending order increases the average PSNR of recovered frames by approximately 3-9 dB for Akiyo, 5-9 dB for Foreman, and 5-9 dB for Coastguard video. In Table 4.4, we also compare our results to some existing methods (40 numbers of hypotheses) for various CIF videos. The average PSNR of video frames is comparable to MH-TIK, MH-wEnet, MH-ST. However, MH-RTIK gives better results than all other methods. In Table 4.5, we compare reconstruction time with some existing methods (40 numbers of hypotheses). Our method takes less time than that all other methods. We consider CR=0.7 for the reference frame and CR lies between 0.1 to 0.5 for non-reference frames. Our proposed method can be used in the compressive distortion minimizing rate control

(C-DMRC) system [68] as it has a less complex encoder and decoder and can withstand considerably higher bit error rates. Traditional video coding systems, such as H.26X/MPEG-X, are unsuitable for applications such as wireless video sensor networks (WVSN) because the encoder design is more difficult than the decoder design [102], and it is constrained by computational resources and storage space. Our proposed method can be suitable for the above-said applications.

We have compared our proposed measurement matrix with the Hadamard measurement matrix for our scheme using different videos in Table 4.6. Our proposed measurement matrix gives a better average PSNR of recovered video than that of the Hadamard measurement matrix. Also, we have compared reconstruction time while considering the Hadamard matrix as the measurement matrix in Table 4.7. Our proposed measurement matrix takes less time for reconstruction than the Hadamard matrix. In our case measurement matrix does not change its value if we will run it multiple times. So the average PSNR of the recovered video frame remains fixed for any iteration. However, in the case of the Hadamard matrix, the rows are selected randomly while sampling the video data. So more iteration is required to get a better average PSNR while reconstructing the video data.

Table 4.3: Average PSNR and reconstruction time for different compression schemes

Methods	**Parameters**	**CR**				
		0.1	**0.2**	**0.3**	**0.4**	**0.5**
Fang et al. [23]	PSNR(dB)	24.17	27.29	30.31	33.75	38.22
Pudlewski et al. [68]		24.43	27.53	29.76	32.4	35.78
Our(GOP=2)		24.51	28.28	30.82	35.05	41.9
Our(GOP=4)		24.46	28.19	30.75	35.08	41.83
Fang et al. [23]	Reconstruction time(sec)	12.85	14.3	20.17	18.4	18.67
Pudlewski et al. [68]		52.32	47.34	37.23	37.08	30.49
Our(GOP=2)		0.93	0.97	2.14	2.22	2.30
Our(GOP=4)		1.53	1.56	2.54	2.76	2.82

4.4.3.2 Performance Analysis of Video Encryption

The mean correlation coefficient for encrypted video frame has to be less for better security. The encrypted frames are highly uncorrelated which indicates our method for video encryption is secure. The mean CC for encrypted GOP frame is nearer to

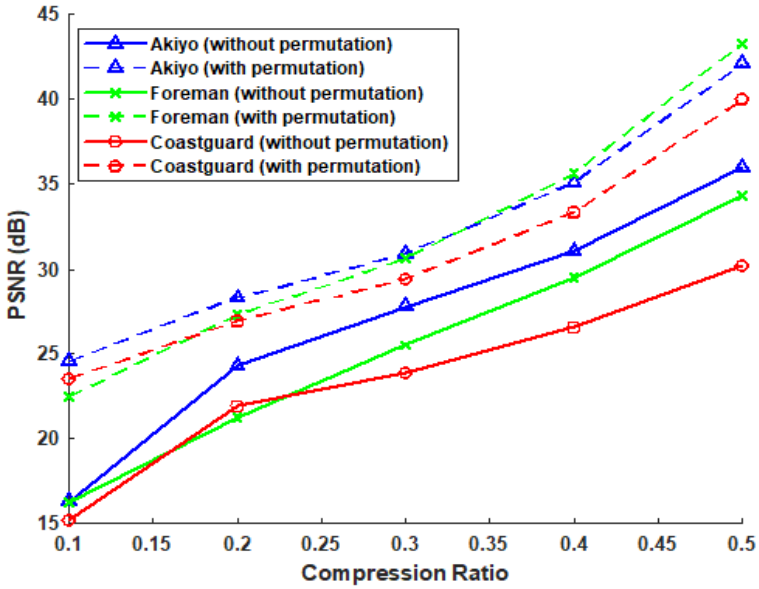


Figure 4.11: Average PSNR of recovered video with and without permutation

Table 4.4: Average PSNR of recovered video using different compression schemes

Video	**CR (Non Ref)**	**Our**		**MH-TIK**	**MH-wEnet**	**MH-ST**	**MH-RTIK**
		GOP=2	**GOP=4**	**GOP=2**			
Foreman	0.1	31.35	29.90	33.91	34.13	34.22	35.64
	0.2	31.73	30.32	34.4	34.5	34.63	36.06
	0.3	32.17	30.83	34.86	35.02	35.09	37.56
	0.4	32.67	31.31	35.31	35.26	35.48	37.96
	0.5	33.16	31.89	35.93	35.21	35.89	38.60
Hall	0.1	34.54	34.32	32.83	33.01	33.30	37.10
	0.2	34.63	34.39	33.44	33.66	33.61	38.01
	0.3	34.71	34.52	33.91	34.08	34.15	38.39
	0.4	34.81	34.63	34.25	34.57	34.62	38.65
	0.5	34.92	34.75	34.75	34.99	35.04	38.91
Coastguard	0.1	30.99	29.77	30.59	30.77	30.94	33.12
	0.2	31.43	30.10	31.35	31.23	31.57	34.26
	0.3	31.87	30.46	31.67	31.64	32.10	34.69
	0.4	32.00	30.86	32.01	32.19	32.49	35.08
	0.5	32.53	31.30	32.41	32.00	33.90	35.43
Container	0.1	31.98	31.33	30.74	29.96	30.54	33.67
	0.2	32.28	31.62	31.28	30.43	31.49	34.76
	0.3	32.46	31.84	31.67	31.31	31.87	35.08
	0.4	32.63	31.94	32.01	31.84	32.33	35.28
	0.5	32.88	32.16	32.45	32.24	32.65	35.47

Table 4.5: Average reconstruction time(sec) for different compression schemes

Video	CR (Non Ref)	Our		MH-TIK	MH-wEnet	MH-ST	MH-RTIK
		GOP=2	GOP=4	GOP=2			
Foreman	0.1	4.4	3.3	7.5	3.8	9.7	17.0
	0.2	4.6	3.5	7.7	5.5	10.1	17.8
	0.3	6.1	5.8	8.2	8.2	10.6	20.3
	0.4	6.3	6.2	8.3	12.7	10.8	20.4
	0.5	6.7	6.8	8.3	20.8	10.9	20.3
Hall	0.1	4.9	3.3	7.3	3.7	9.5	16.9
	0.2	5.1	3.7	7.5	5.5	9.7	17.2
	0.3	6.3	5.6	7.9	7.9	10.3	19.9
	0.4	6.6	6.3	8.0	11.7	10.4	20.1
	0.5	7.1	6.7	8.2	19.9	10.7	20.2
Coastguard	0.1	4.4	3.4	7.7	4.1	10.0	17.5
	0.2	4.5	3.6	7.9	5.8	10.2	17.6
	0.3	6.4	5.7	8.4	8.5	10.8	20.5
	0.4	6.7	6.3	8.5	12.3	11.0	20.6
	0.5	6.9	6.8	8.5	20.2	11.2	20.7
Container	0.1	4.7	3.3	7.6	3.9	9.9	16.9
	0.2	4.9	3.6	7.6	5.6	10.1	17.3
	0.3	6.2	5.8	8.1	8.0	10.5	20.1
	0.4	6.6	6.3	8.3	12.1	10.8	20.4
	0.5	7.0	6.7	8.2	20.5	11.1	20.4

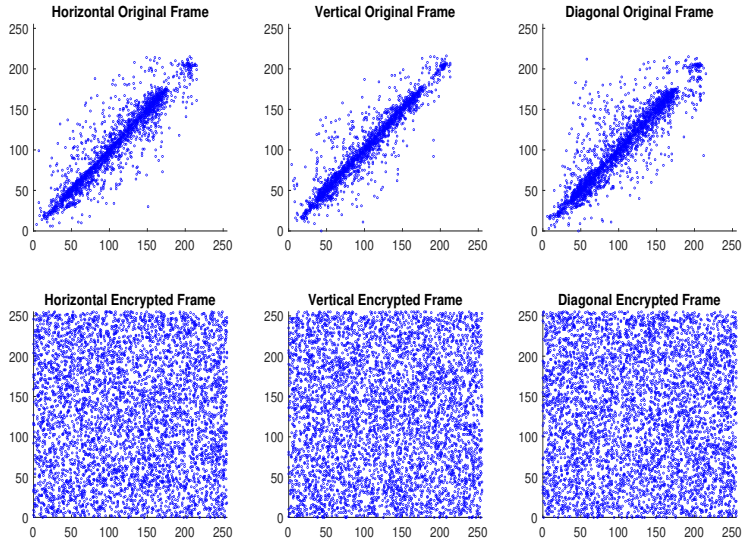


Figure 4.12: Correlation distribution of Akiyo video (a) Horizontal, (b) Vertical, (c) Diagonal and (d-f) its encrypted frame

Table 4.6: Average PSNR of reconstructed video for different measurement matrices

Video	CR	Measurement matrix			
		Hadamard		LT-ICM	
		GOP=2	GOP=4	GOP=2	GOP=4
Akiyo	0.1	22.48	21.71	24.51	24.46
	0.2	26.8	25.99	28.28	28.19
	0.3	29.06	28.68	30.82	30.75
	0.4	33.53	33.57	35.05	35.08
	0.5	40.19	39.43	41.9	41.83
Coastguard	0.1	23.48	23.02	23.78	23.56
	0.2	25.76	25.62	26.99	26.49
	0.3	28.73	28.59	29.58	28.85
	0.4	32.65	31.11	33.13	32.02
	0.5	39.91	38.14	40.34	39.98
Foreman	0.1	21.95	20.53	23.28	22.82
	0.2	26.32	25.26	26.9	26.07
	0.3	28.91	27.94	30.14	28.76
	0.4	32.87	31.18	34.96	32.58
	0.5	39.92	38.56	42.39	41.1

Table 4.7: Average reconstruction time (sec) for different measurement matrices

Video	CR	Measurement matrix	
		Hadamard	LT-ICM
Akiyo	0.1	2.36	0.93
	0.2	2.69	0.97
	0.3	3.32	2.14
	0.4	3.77	2.22
	0.5	4.03	2.3
Coastguard	0.1	3.51	1.06
	0.2	3.63	1.2
	0.3	3.82	2.2
	0.4	4.6	2.39
	0.5	6.38	2.02
Foreman	0.1	3.17	1.03
	0.2	3.5	1.19
	0.3	3.76	2.33
	0.4	3.89	2.53
	0.5	4.16	2.03

zero for our method. We compare mean CC and entropy for the encrypted video frame using our proposed method with some other existing methods [58] [83] [55] [85] [87] [89] [96] [91] [101]. We observed that encrypted frames using our proposed method have less mean correlation coefficient. The distribution of adjacent pixels for original and encrypted GOP frame number 3 of Akiyo video in all directions for Y component is shown in Fig. 4.12. The CC can be found between two pixels (m,n) in a frame using following equation:

$$CC_{mn} = \frac{cov(m,n)}{\sqrt{D(m) \times D(n)}} \tag{4.15}$$

where $cov(m,n)$ is the covariance between m and n. $D(m)$ and $D(n)$ are the variances of m and n respectively.

The entropy of encrypted frames need to be high for unpredictability. Our proposed method gives entropy close to 8 which shows that the encrypted video is unpredictable. Our method has good performance in terms of mean CC and entropy of encrypted video frames which are shown in Table 4.8. The formula for calculating the entropy H(f) of a source is as follows:

$$H(f) = -\sum_{i=1}^{255} P(f_i) \log_2 P(f_i) \tag{4.16}$$

where $P(f_i)$ is the probability of symbol f_i.

To protect against differential attacks, the NPCR and UACI metrics should exceed the benchmark values of 99.5693 and 33.4635, respectively [103]. The mathematical formulations of NPCR and UACI are given as follows:

$$NPCR(P_1P_2) = \sum_{i,j} \frac{F(i,j)}{M \times N} \times 100\% \tag{4.17}$$

$$UACI(P_1, P_2) = \sum_{i,j} \frac{|P_1(i,j) - P_2(i,j)|}{255 \times M \times N} \times 100\% \tag{4.18}$$

where P_1, P_2 denote two encrypted frames obtained from the same video frame and key, differing by only one pixel. Let $M \times N$ represent the frame dimensions, and $F(i,j) = 1$ if $P_1 \neq P_2$ and is 0 otherwise 0. Utilizing [56], we conducted exper-

iments to validate the resistance of our scheme against differential attacks. Various cases were considered, such as altering a single pixel in the source frame, modifying the initial conditions, and adjusting the control parameters. In all situations, the proposed method achieves NPCR and UACI values above the standard thresholds, ensuring strong robustness against differential attacks. The decrypted video frames for different videos with different CR values are shown in Fig. 4.13.

We compare the encryption time of our approach with that of other current methods [87] [104] for various file sizes. Our proposed method takes significantly less time for encryption than the other two methods. This is shown in Table 4.9

Table 4.8: Mean CC, Entropy, NPCR and UACI for different schemes

Methods	**CC after Encryption**			**Entropy**	**NPCR**	**UACI**
	Horizontal	Vertical	Diagonal			
Sethi et al. [58]	-0.0020	-0.0048	-0.0114	7.9993	99.62	33.48
Unde et al. [83]	0.0030	0.0027	0.0014	7.9900	99.69	33.55
Ranjithkumar et al. [55]	-0.0372	0.0246	0.0313	-	99.55	33.55
Song et al. [85]	-0.0245	-0.0135	-0.0143	7.8708	-	-
El-Latif et al. [87]	-0.0012	0.0004	-0.0003	7.9984	99.62	-
Muhammad et al. [89]	0.0031	0.0026	0.0006	7.9900	99.61	33.44
Karmakar et al. [96]	-0.0023	0.0069	0.0067	7.9985	99.62	33.47
Karmakar et al. [91]	-0.0010	0.0057	0.0055	7.9987	99.61	33.48
Liu et al. [101]	0.0090	0.0091	0.0098	7.9981	99.6	33.51
Our Method	-0.0005	-0.0012	-0.0001	7.9989	99.64	33.57

- indicate data are not available

Table 4.9: Encryption time (sec) for different schemes

Methods	**CR**	**Data Size**			
		100KB	**1MB**	**10MB**	**100MB**
Our	0.1	0.0599	0.0986	0.4553	4.9851
	0.2	0.0674	0.2931	0.7421	9.0794
	0.3	0.0752	0.1635	1.0183	13.2716
	0.4	0.0802	0.1919	1.2953	17.5552
	0.5	0.1102	0.2236	1.563	21.8336
El-Latif et al. [87]		0.194	2.0845	21.9606	250.3024
Yang et al. [104]		9.2	42.8	361.7	4487.5

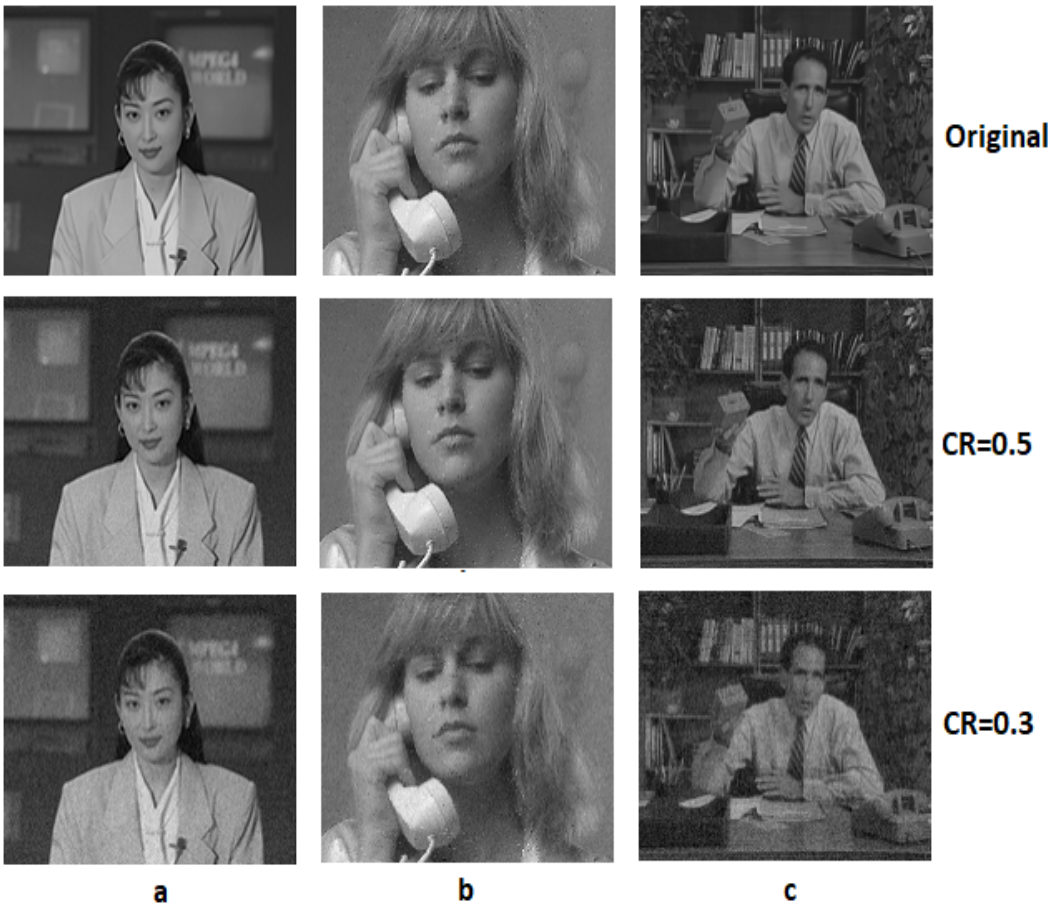


Figure 4.13: Quality of different recovered video frames for two CR values for three different video - (a) Akiyo (b) Suzie and (c) Saleman

4.5 Discussion

In this work, we propose a fast method for joint compression and encryption of video frames. The proposed method considers new permutations before applying parallel compressive sensing using an improved chaotic map. The new permutation enhances the average quality of the recovered video. Our proposed substitution method and data-dependent chaotic sequence generation which is used for shuffling make video frames more secure. Experimental results and comparisons with state-of-the-art methods clearly demonstrate the efficacy of our solution. We plan to use different video codecs [105] considering motion part in the future for our joint compression and encryption strategy. Another direction of future research will be to explore different joint reconstruction algorithms for better recovery of video.

Chapter 5

A Robust and Secure Video Recovery Scheme with Deep CS

This chapter introduces a secure, high-quality video recovery scheme for telemedicine and cloud-based monitoring. After implementing deep learning-based video Compressive Sensing (CS), we encrypt the compressed video. We divide a video into GOPs with keyframes and non-keyframes. The suggested video CS approach employs a CNN with an SSIM-based loss function. Our recovery process consists of two stages. The first recovery step uses CNN to optimise spatial redundancy. Keyframes and neighbouring non-keyframes compensate for non-keyframes in deep recovery. We use multilevel feature compensation for keyframes and single-level compensation for neighbouring non-keyframes. We also propose a Sine Symbolic Chaotic Map (SSCM), an unpredictable and complex chaotic map with a wider chaotic range. We propose a safe encryption approach for compressed features using Forward Diffusion, Substitution, Backward Diffusion, and XORing with SSCM-based chaotic sequence. We demonstrate the superiority of our combined approach over many state-of-the-art image and video CS algorithms and video encryption methods via extensive testing.

5.1 Introduction

In recent years, video compression and encryption have received considerable attention. In this work, we have addressed video compression and encryption in a combined manner with the overall goal of achieving secure, high-quality video recovery with low computational costs. Our solution can be helpful for practical applications that require high compression ratio and confidentiality, such as telemedicine [64] and cloud-based surveillance [65]. Ensuring the security of large amounts of sensitive

medical video data transmitted to the cloud is of utmost importance in telemedicine. So, our proposed scheme can be suitable for both applications.

A common problem in sole video encryption techniques is that both the input and the encrypted video are of the same size. This opens the door for the use of compression methods, among which Compressive Sensing (CS) strategies have become popular. Video compression with different video codecs [66, 67] has a complex encoder which requires more computational power to process the video data than the decoder. So, such schemes are not well suited for low-power applications. On the other hand, the CS concept which deals with sampling and representing a video signal with a few non-zero coefficients is required to form a simple encoder. In compressive video sensing (CVS), compression and sampling are performed simultaneously, which is desirable for many imaging applications [68, 69]. Two methods [70, 71] use image-based CS while recovering compressed frames. These methods treat each video frame as an image. Image-based CS methods provide low-quality recovered video frames because they ignore temporal correlation. Different CVS methods [72, 73] employ both spatial and temporal correlation while recovering the complete video by considering the whole video sequence as a volume. However, these methods have high computational complexity [69]. Most CVS methods ([24, 74, 75]) perform frame-wise initial reconstruction using the measurements and then perform motion estimation and compensation to improve current frame reconstruction.

There are several important differences between the previous conference version [106] and this current journal version. In [106], we suggested a deep learning-based video compression scheme that is capable of recovering high-quality video with a reasonable recovery time compared to conventional CVS schemes. In this work, we present a robust and secure combined video compression and encryption scheme at the GOP level, where, compression is performed using deep CS. This is followed by an encryption process using Diffusion Substitution Diffusion XORing (DSDX). Our proposed encryption approach uses MixColumns, ShiftRows of AES [107], and XOR operations for forward and backward diffusion. Secondly, we optimize the measurement matrix using the convolution layer as opposed to a content-independent measurement matrix. Thirdly, we propose a robust technique to encrypt the GOP-level compressed data using diffusion and substitution processes with an SSCM chaotic sequence. Fourthly, we have shown in this manuscript how the quality of the recovered video has improved even at lower sampling ratios (SR). Finally, we have incorporated extensive experimentation using six standard CIF video sequences. Our key contributions are summarized as follows.

1. We propose a GOP level scheme for video compression and encryption. The proposed scheme is secure, recovers good quality video, and exhibits recovery time comparable to traditional CVS schemes.

2. Our SSIM-based deep CS scheme compensates non-keyframes from both keyframes and neighboring non-keyframes. In a single keyframe-based recovery scheme, one keyframe and nearby non-keyframes compensate non-keyframes in a GOP. In a double keyframe-based scheme, two keyframes provide bidirectional compensation for the non-keyframes, in addition to the compensation from neighboring non-keyframes. These methods recover high-quality video even at very low sampling ratios.

3. We suggest a new chaotic map that takes into account two existing chaotic maps, both of which have a less chaotic range. The suggested map increases chaotic range, complexity, and unpredictability which makes it very effective for video encryption.

4. We present an encryption scheme for the compressed data. The proposed encryption scheme is robust, and is able to defend against statistical, differential, and cropping attacks. Encrypted video for the proposed scheme has low correlation coefficients (CC), high entropy, and a high number of changing pixel rate (NPCR).

.

The rest of the work is organized as follows: In Section 5.2, we discuss the related works and point out their limitations. In Section 5.3, we describe our proposed video compression in detail. Section 5.4 presents the experimental results with analysis. Finally, the work has a summary in Section 5.5.

5.2 Related Work

We divide this section into three subsections. In the first subsection, we discuss different existing image and video compressive schemes. In the second subsection, we discuss various methods for image and video encryption. In the last subsection, we discuss some combined video compression and encryption schemes.

5.2.1 Compressive Image and Video Sensing

We proposed a combined compression and encryption scheme for recovering high-quality video frames that incorporates deep compressive sensing and a new encryption pipeline. We combine our previous CVS compression framework discussed in Chapter 3 with the newly developed encryption methodology. We discussed different image and video compressive sensing schemes in 3.2 of Chapter 3.

5.2.2 Video Encryption

This section discusses some of the most notable research on CS-based cryptosystems. Using stream cipher, the authors in [34] proposed a CS-based video encryption method. However, their technique is appropriate for higher sampling ratios. The authors of [2] proposed a video compression and encryption scheme that did not consider temporal redundancy within frames. In their encryption process, they employed a 5D hyperchaotic system and a DNA encoding scheme. The use of patch-based operation causes the blocking effect on the reconstructed frames. Different video encryption techniques using chaotic maps are discussed in [9, 13, 58, 85]. Controlled XOR operations are used with an enhanced logistic map in a video encryption technique suggested in [85]. The updated logistic map provides a wider variety of chaotic possibilities than the original logistic map. However, the chaotic range is limited for this method [85]. Chaotic maps used for video encryption in [13] are not secure for a wide chaotic range. Sethi *et al.* [9] proposed a new one-dimensional chaotic map Logistic Tent Infinite Collapse Map (LT-ICM) which has a higher chaotic range and complex behavior, unpredictability, and sensitivity towards initial values and controlling parameters. There is an effective method described in [89] that uses probabilistic encryption to protect video summary keyframes. Quantum walks-based substitution boxes are used for video data encryption in [87]. Frame-level video encryption has been presented in [58] by authors employing the logistic sine coupling map (LSCM), which maintains an interframe relation while creating the chaotic sequence. However, the method discussed in [87], [58], [13] did not consider temporal redundancy while encrypting the video. The chaotic maps used for encryption methods [58,85,89] have a smaller chaotic range. It is preferable to have a map with a high degree of chaos and with a high chaotic range. We use SSCM-based chaotic sequence in our encryption system.

In [108] a novel fractional order discrete improved Hénon map (FODIHM) and deoxyribonucleic acid (DNA) sequences are used to propose a color image encryp-

tion scheme. Rubik's cube transformation has been applied to every channel in the image during the permutation stage. Moreover, many DNA coding and CAT transformation algorithms, along with the Fractional Order (FO) discrete logistic map and XOR operator, have been used in the scrambling step to modify the position of pixels in every color channel. It is shown that the presented scheme is resistant to known attacks using various analyses. In [109], the authors presented a symmetric image encryption technique based on the Lorenz chaotic system. The method achieved double scrambling by utilizing two keystreams for pixel-level scrambling and a cyclic shift operation for bit-level scrambling. The double scrambling process improves security. An improved diffusion process encrypts the image using a separate keystream. Their scheme employs a unique keystream generation method. However, our method gives lower correlation coefficients and higher NPCR, UACI, and entropy of encrypted frames than methods presented in [108, 109].

5.2.3 Integrated Video Compression and Encryption

Now we discuss a scheme that integrates video compression with encryption. Sethi *et al.* [4] proposed a joint video compression and encryption scheme at the GOP level using PCS and LT-ICM. A content-independent chaotic map based on LT-ICM is used in method [4] to construct the measurement matrix which cannot provide high-recovery quality video at lower sampling ratios such as 0.01 and 0.05. Also, the traditional CVS reconstruction techniques, such as orthogonal matching pursuit (OMP) and basis pursuit are impacted by signal sparsity, measurement matrix size, and noise sensitivity. As a consequence, traditional CVS schemes fail to give good quality recovery results. The encryption scheme [4] used two rounds of operation to attain the desired value of different security parameters. Since our method is based on deep learning, compressed measurements are optimized during the training phase to produce high-quality video with a low SR. To get high-quality recovered video, non-keyframes that are sampled at low SR are compensated from both keyframes in multi-level and surrounding non-keyframes in single-level using SSIM-based loss function. Our proposed encryption method consists of forward diffusion, substitution, backward diffusion and finally XORing operation with the SSCM chaotic sequence. The forward and backward diffusion processes are used to avoid multiple rounds and to ensure a high NPCR value. The substitution operation is performed to achieve the required confusion. Since three operations namely MixCloumns, SubBytes, and ShiftRows are key independent, so final XOR-ing with key-dependent chaotic sequence is employed to make the proposed encryption scheme

secure. In [110], the authors developed a retrieval image compression and encryption (RICE) technique for cloud computing applications that offers a balanced solution for retrieval, compression, and encryption. Our suggested scheme performs better in terms of different encryption parameters than the one employed in [110] as it has higher NPCR, UACI, entropy, and lower correlation coefficients.

Our combined solution consists of deep learning-based video Compressive Sensing (CS) followed by novel video encryption. The proposed combined scheme exhibits superior compression efficiency as our scheme can give good recover quality video at very low sampling ratios like 0.01 and 0.05 which is not possible for traditional VCS methods. Our scheme can provide good security as the encrypted video frame has high NPCR, UACI, and lower mean correlation coefficients. Video compression using CS and encryption of compressed data can be achieved in three ways - (a) By using classical solutions for both CS and encryption, (b) By applying a deep learning-based CS but classical encryption method and (c) By adopting deep learning based solutions for both CS and encryption. If both video CS and encryption of CS data are performed using classical methods, good recovery quality of video cannot be ensured at lower sampling ratios like 0.01 and 0.05. In contrast, if both the video CS and encryption are performed using deep learning-based approaches, then better recovery results are obtained at the cost of very high complexity. Treating Video CS using a deep learning method will provide good recovery quality video at lower sampling ratios, and performing encryption using the classical method will incur lesser computational costs. So in our proposed method, we perform video CS using a deep learning-based method and encryption of compressed features using the classical DSDX method. AddRoundKey, MixColumns, SubByte, and ShiftRorws are the four basic phases of the AES algorithm. To ensure strong security in AES, each step requires many rounds. For example, with a 128-bit key, each step in AES takes 10 rounds. The encryption and decryption process [111] takes around 14 minutes for 512-byte data. However, our proposed encryption and decryption scheme for single keyframe-based network takes only 8 seconds with, 934400-byte data. The encryption and decryption time is less due to the architecture of the encryption and decryption scheme and fewer numbers of rounds in encryption and decryption. Similarly, for double keyframe-based encryption and decryption scheme takes around 15 seconds for 1753600 bytes of data. Bidirectional diffusion employing Mixcolumns, ShiftRows, and Xor operations is what we propose to prevent several rounds of diffusion. The encryption scheme [4] used two rounds of operation to get the desired security parameter. The GOP size in method [4] is 2 and 4. If we

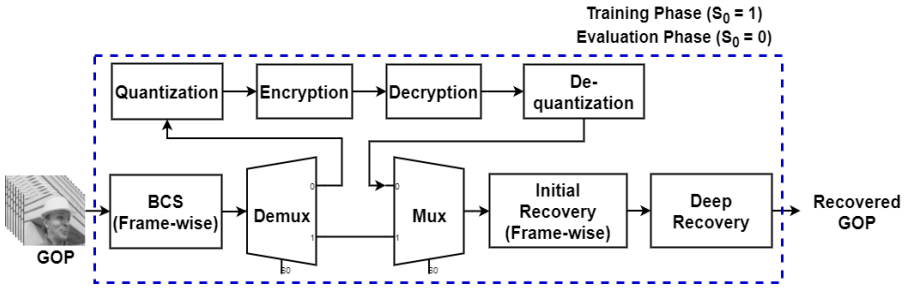


Figure 5.1: A block diagram of the proposed combined video compression and encryption scheme

increase the GOP size, the recovery result decreases, as the difference between two consecutive frames affects the sparsity of the signal. Also, our suggested encryption scheme has a higher NPCR, UACI, and low correlation coefficients than the method used in [4].

The literature survey shows that there are mostly separate schemes for video compression and video encryption. We also note that relatively few strategies for encrypting and compressing video using CS have been proposed. However, to the best of our knowledge, there is no video compression and encryption scheme available where both deep learning and conventional (without deep learning) strategies are combined. In this work, we present how deep learning-based CVS and DSDX-based encryption can be used in a combined way to design a robust and secure video compression and encryption scheme.

5.3 Proposed method

We have already discussed our two methods (Single keyframe and double keyframe) in 3.3. The first section describes the generation of the proposed chaotic map and its evaluation. In the last subsection, we explain the encryption of compressed data using a new DSDX-based approach. The complete block diagram of the proposed method is shown in Fig. 5.1. When the selection input S_0 of Mux and Demux is 1 then, the model will operate in the training phase to optimize the compressed features for better recovery. In the evaluation phase, Demux selects the compressed features for quantization, followed by encryption. After decryption and de-quantization of the compressed features, the signal is supplied to the initial recovery network, followed by the deep recovery network through Mux.

Algorithm 9 Video compression and recovery process in CVS

Input: Original video
Output: Recovered video

1: Apply BCS to each keyframe and non-keyframe in a GOP using Eq.(3.2) and Eq. (3.3) respectively
▷ /* Initial Recovery */
2: Recover compressed keyframe and non-keyframe blocks using Eq. (3.4) and Eq. (3.5) respectively, then combine all blocks per frame for frame-level initial recovery
▷ /* Deep Recovery */
3: Extract features from initially recovered keyframe using Eq.(3.6) in stage 1
4: Compensate the features at i^{th} stage with the help of features at $(i-1)^{th}$ stage using Eq.(3.7)
5: Repeat step 4 upto $(n-1)^{th}$ stage
6: Convert feature level data to frame level at n^{th} stage using Eq. (3.8)
7: Extract features from each non-keyframe using Eq.(3.9) in stage 1
8: Compensate non-keyframes features at i^{th} stage with the help of non-keyframe features at $(i-1)^{th}$ stage and keyframe features at i^{th} stage using Eq.(3.10)
9: Repeat step 8 upto $(n-2)^{th}$ stage
10: Compensate features at $(n-1)^{th}$ stage with the help of non-keyframe features at $(n-2)^{th}$ stage, keyframe features at $(n-1)^{th}$ stage, and neighboring non-keyframe features at $(n-1)^{th}$ stage using Eq.(3.11)
11: Finally, transform feature domain data to frame level data at the n^{th} stage using Eq.(3.12)

5.3.1 Chaotic sequence generation

Many one-dimensional chaotic maps, such as sine [112] and symbolic maps [113], are characterized by a low chaos range. The symbolic map is expressed as:

$$x_{n+1} = -a_1 \times x_n + \frac{x_n}{|x_n|} \tag{19}$$

where a_1 is the controlling parameter, and it behaves chaotically for $a_1 \in [0, 2]$.

The sine map is expressed as:

$$x_{n+1} = a_2 \times sin(\pi \times x_n) \tag{5.1}$$

where a_2 is the controlling parameter, and it behaves chaotically when $a_1 \in [0.87, 1]$ [112].

We present a novel chaotic map that integrates the Sine and symbolic maps, allowing it to exhibit chaotic behavior throughout an expanded range of the controlling

parameter. The suggested map is articulated as:

$$x_{n+1} = mod(a_3 \times sin((-20 + a_3) \times \pi \times x_n + \frac{\pi \times x_n}{|\pi \times x_n|}, 1) \tag{5.2}$$

where a_3 is the controlling parameter and $a_3 \in [0, 100]$.

5.3.1.1 Evaluation of proposed chaotic map

We examined the proposed chaotic map considering the bifurcation diagram, Permutation entropy, and Lyapunov exponent. A bifurcation diagram is a useful tool for describing the dynamic behavior of chaotic maps. The attractor must occupy the larger areas in phase space for excellent chaotic performance. We consider 0.765 as the initial state and iterate Sine, Symbolic, and proposed map 2000 times. In Fig. 5.2, we show the bifurcation diagrams of each chaotic map and found that the attractor covers most of the areas of phase space for our proposed map.

Lyapunov exponent (LE) defines the rate at which two adjacent trajectories differ from one another after a significant number of iterations. It is an important parameter that characterizes the chaotic behavior of any chaotic map. The mathematical expression for LE for a 1D chaotic map $x_{n+1} = f(x_n)$ can be expressed as:

$$LE = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} ln \left| f^1(x_k) \right| \tag{5.3}$$

The LE value should be high and positive for high unpredictability. Fig. 5.3 confirms that our suggested chaotic map has a greater LE than the sine and symbolic map.

The complexity of a chaotic system can be measured by permutation entropy (PE) and for a better unpredictable chaos system, the PE value has to be higher. We use method [114, 115] for measuring the PE of the chaotic sequence generated from sine, symbolic, and our proposed chaotic map. In Fig. 5.4, we show the PE for the aforementioned chaotic maps and found that the PE for our proposed chaotic map is nearer to 1, indicating the generated sequence is unpredictable for the entire range of controlling parameters.

NIST SP 800-22 is a popular random number generator test suite. It assesses statistical randomness with 15 subtests. If all SP 800-22 subtests pass, the generated pseudorandom sequence is statistically random. The P value has to be greater than 0.01 to pass the subtests [116]. We generate 10^7 numbers of bits considering the

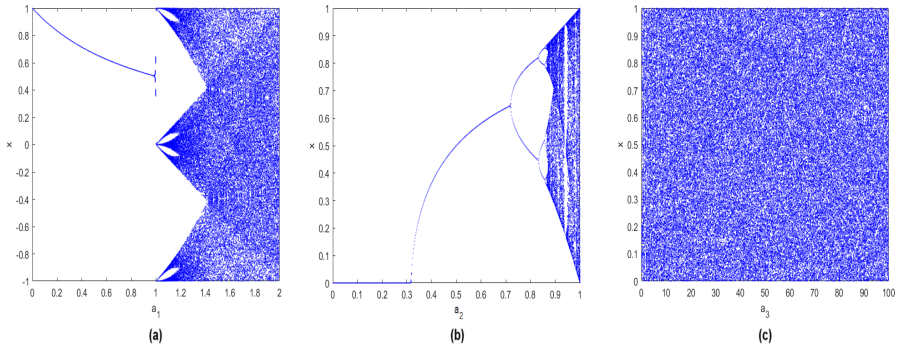


Figure 5.2: Bifurcation diagram for (a) Symbolic Map (b) Sine map (c) proposed map

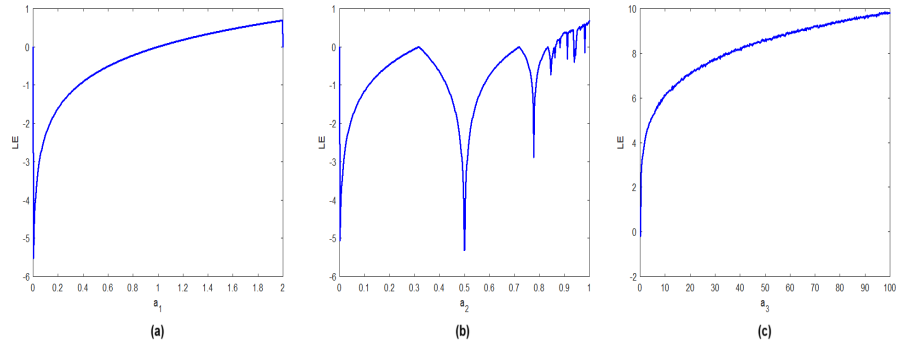


Figure 5.3: Lyaponov exponent for (a) Symbolic Map (b) Sine map (c) proposed map

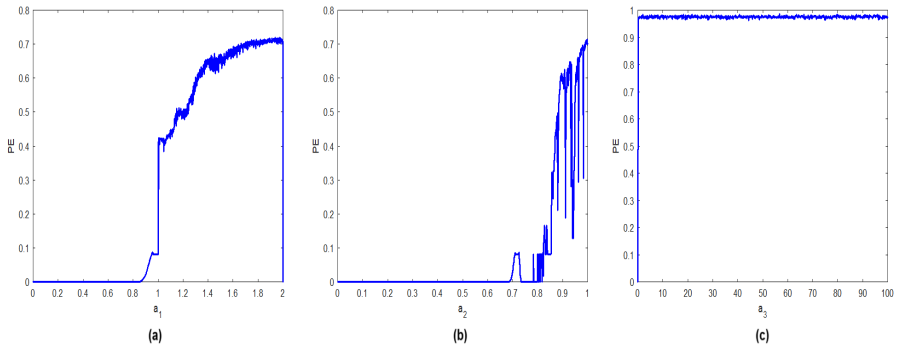


Figure 5.4: Permutation entropy for (a) Symbolic Map (b) Sine map (c) proposed map

controlling parameters and initial states. We show in Table.5.1 that all of the subtests have p-values greater than 0.01, which means that the pseudorandom sequence generated from our proposed chaotic map passes all of them.

5.3.2 Encryption of compressed video

This subsection covers how our suggested method encrypts compressed frames. A compressed frame (C) is quantized before encryption. Eq.(5.4) is used to quantize

Table 5.1: Randomness test for proposed chaotic map

Sub-tests	P-value (>0.01)	Status
Frequency	0.654443	P
Block frequency	0.305447	P
Cumulative Sums	0.600072	P
Runs test	0.800948	P
Longest run	0.824689	P
Binary matrix rank	0.367088	P
FFT	0.621784	P
Non-overlapping template	0.998131	P
Overlapping template	0.572090	P
Universal	0.405121	P
Approximate entropy	0.985659	P
Random excursions	0.262846	P
Random excursions variant*	0.933604	P
Serial	0.561047	P
Linear Complexity	0.503955	P

P indicate Pass

compressed frames, and the inverse of the quantization process is performed using Eq.(5.5).

$$C_Q = round(((C - C_{min}) \times 255)/(C_{max} - C_{min})) \tag{5.4}$$

$$C = (((C_{max} - C_{min}) \times C_Q)/255) + C_{min} \tag{5.5}$$

where C_Q is the quantized version of compressed frame C and C_{min} and C_{max} are the minimum and maximum values of C. We suggest a new architecture (DSDX) for encrypting compressed video features utilizing chaotic sequence-based XORing and AES-based MixColumns, ShiftRows, and SubBytes. Our encryption approach uses MixColumns and ShiftRows operations of AES [107] for forward and backward diffusion. To achieve the effect of diffusion over all blocks, the previous diffused block is XOR-ed with the present block before processing further. We employ substitution using the AES substitution box between the two diffusion processes to create confusion. Finally, the encrypted output is produced by XORing the result of the second-level diffusion step with a chaotic sequence generated using the proposed SSCM. The detailed procedure for the encryption of compressed data is given in Algorithm-10. In our method, we encrypt compressed keyframes separately and all non-keyframes together in a GOP. The block diagram of our proposed encryption

Algorithm 10 Encryption of compressed data

Input: Compressed video data, initial states, and controlling parameters
Output: Encrypted frames

1: Divide quantized compressed keyframe into n blocks (B_1, B_2,..., B_n) each of size 4×4 bytes
2: Generate a SSCM-based chaotic sequence using initial states and controlling parameters and divide it into (n+2) blocks (S_F, S_1, S_2,...., S_n, S_B) each of size 4×4 bytes
3: For the first block, perform XOR operation between B_1 and S_F, then pass through AES-based MixColumns module
4: For blocks B_2 to B_n, perform XOR between the current block and the previous diffused block output obtained after passing step-3 output through AES-based ShiftRows module
5: Perform substitution at byte level on MixColumns output using SubBytes operation of AES algorithm
6: Perform steps 3 and 4, starting from the last processed block obtained after substitution operation and chaotic sequence block S_B
7: Perform XOR operation between the processed blocks obtained from step 6 and the chaotic sequence blocks (S_1, S_2,...., S_n)
8: Combine all the processed blocks to get the encrypted keyframe
9: Apply steps 1 to step 8 for encrypting all non-keyframes in a GOP at once

module is shown in Fig. 5.5.

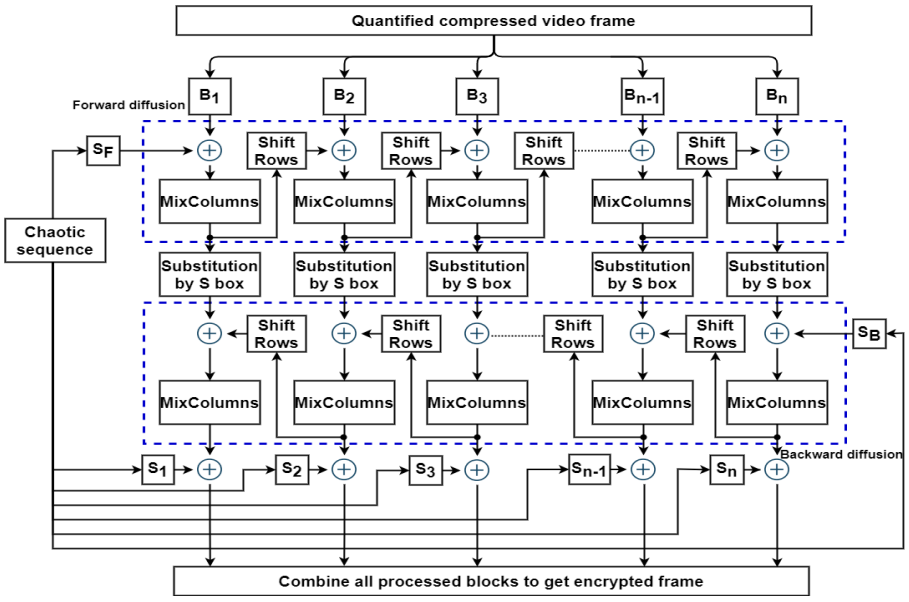


Figure 5.5: Block diagram of DSDX-based encryption of compressed data

5.4 Experimental Analysis

In this section, we describe our experiments in detail and show the efficacy of our proposed method through a thorough analysis. All experiments are carried out on a desktop PC with Intel®Xeon(R) CPU E5-2690 v4 @ 2.60GHz, 16 Core with NVIDIA Quadro K2200 GPU. The proposed scheme is implemented using MATLAB R2020a. To test our model, we have used six test video sequences[1], namely, Akiyo, Coastguard, Foreman, Mother_daughter, Paris, and Silent. Results are compared with several state-of-the-art deep learning-based image CS methods ([29], [76], [30]) and video CS methods ([24], [75], [74], [4] and [20]). We also compare our performance with some existing encryption methods [2, 4, 13, 34, 58, 85, 87, 89].

5.4.1 Comparisons with SOTA Image and Video CS methods

We have already shown some qualitative visual quality comparisons of different image and video CS methods on 2^{nd} frame of Akiyo video for sampling ratio = 0.1, along with rate-distortion performance comparison for different video compressive sensing methods in 3.4.3. Furthermore, in Table 5.2, we compare the average PSNR and SSIM of recovered video frames with two approaches of [117] for two videos entitled Crew and City. Our method shows improvement in PSNR and SSIM compared to [117]. Table 5.3 shows the average PSNR and SSIM of recovered video frames for two GOPs when only decompression is used and when both decryption and decompression are used. The decrypted and decompressed results are only slightly affected by the quantization process.

Table 5.2: Comparison of different video CS methods in terms of average PSNR and SSIM of recovered frames

Video	SR=0.0625				SR=0.05			
	FC7-10M [117]		FC7-10M+MMLE [117]		Our (Single Keyframe)		Our (Double Keyframe)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Crew	32.48	0.8771	33.35	0.8943	33.77	0.8976	34.08	0.8988
City	23.29	0.6279	23.16	0.6408	27.03	0.6299	27.24	0.6537

[1]Test videos can be found at https://media.xiph.org/video/derf/

Table 5.3: Comparison of decompressed only and both decrypted and decompressed results in terms of average PSNR and SSIM

Video	SR	Single Keyframe				Double Keyframe			
		Decompress		Decrypt and Decompress		Decompress		Decrypt and Decompress	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Akiyo		34.41	0.9743	34.39	0.9610	40.54	0.9852	40.30	0.9817
Coastguard		27.60	0.6659	27.52	0.6631	28.35	0.6777	27.99	0.6718
Foreman		29.13	0.8233	29.12	0.8231	29.74	0.8325	29.54	0.8323
Mother_daughler	0.01	35.98	0.9367	35.93	0.9262	39.97	0.9604	39.41	0.9590
Paris		24.91	0.8191	24.85	0.8140	26.99	0.8701	26.89	0.8694
Silent		31.22	0.8668	31.01	0.8669	35.10	0.9346	34.74	0.9288
Average		30.54	0.8477	30.47	0.8424	33.45	0.8767	33.14	0.8739
Akiyo		39.15	0.9757	38.99	0.9732	41.51	0.9831	41.03	0.9809
Coastguard		28.99	0.7499	28.98	0.7281	29.77	0.7327	29.19	0.7290
Foreman		32.67	0.8945	32.66	0.8943	33.46	0.9000	33.01	0.8976
Mother_daughter	0.05	40.09	0.9585	39.93	0.9580	40.97	0.9641	40.50	0.9618
Paris		26.95	0.8731	26.94	0.8703	28.23	0.8950	27.94	0.8923
Silent		34.88	0.9292	34.74	0.9239	36.56	0.9473	35.78	0.9362
Average		33.79	0.8968	33.71	0.8913	35.08	0.9037	34.58	0.8996
Akiyo		41.23	0.9795	41.16	0.9791	41.97	0.9838	41.39	0.9799
Coastguard		30.15	0.7756	30.12	0.7752	30.59	0.7898	30.18	0.7821
Foreman		34.26	0.9198	34.14	0.9196	34.74	0.9243	34.00	0.9164
Mother_daughter	0.1	41.53	0.9675	41.44	0.9673	41.65	0.9686	41.10	0.9652
Paris		27.94	0.9055	27.90	0.9022	28.88	0.9079	28.33	0.9000
Silent		36.14	0.9398	36.11	0.9395	36.97	0.9487	36.16	0.9399
Average		35.21	0.9146	35.15	0.9138	35.80	0.9205	35.19	0.9139
Average		33.18	0.8864	33.11	0.8825	34.78	0.9003	34.30	0.8958

5.4.2 Comparisons with SOTA Video Encryption Methods

In Table 5.4, we compare mean CC, entropy, NPCR, UACI of encrypted video frames and key space of our proposed method with state-of-the-art video encryption methods like [2, 4, 13, 34, 58, 85, 87, 89]. For better security, the correlation coefficient (CC) of encrypted video frames should be lower. Encrypted frames must have entropy close to 8 to be unpredictable. To withstand differential attacks, the NPCR and UACI metrics should exceed the standard benchmark values of 99.5693 and 33.4635, respectively [118]. We found that encrypted frames produced by our suggested method have a low mean correlation coefficient, which indicates that the encrypted frames are significantly uncorrelated. The entropy of our suggested method is close to 8, indicating that the encrypted video is unpredictable. The histogram of encrypted keyframe and non-keyframes have uniform distribution. So our proposed encryption scheme can defend against statistical attacks. Our method generates NPCR and UACI values that are above the threshold values, providing a high degree of protection against differential attacks. Our proposed method considers 6

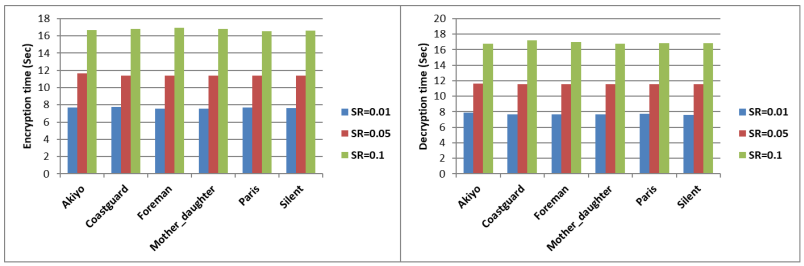
parameters for generating the chaotic sequence. Each parameter has a precision of 10^{14}. So the total key space is $10^{14\times4} = 10^{56} = 2^{186}$. As a result, our method is not susceptible to brute-force attacks. The best values for Table 5.4 are shown in bold. In general, multiple rounds of diffusion and substitution operations are required to achieve better security while encrypting image and video frames. However, in our encryption method, we apply a substitution operation between two diffusion operations, and in the final stage, we apply XOR operation between the second diffusion module output and the chaotic sequence. Fig.5.6 shows encryption and decryption time for our proposed single and double keyframe methods respectively for the first two GOPs of six CIF video sequences. Our proposed scheme takes less time for encryption and decryption. Our suggested approach may be appropriate for situations where fast data encryption is required.

Table 5.4: Comparison of Mean CC, NPCR, UACI, Entropy and key space for different schemes

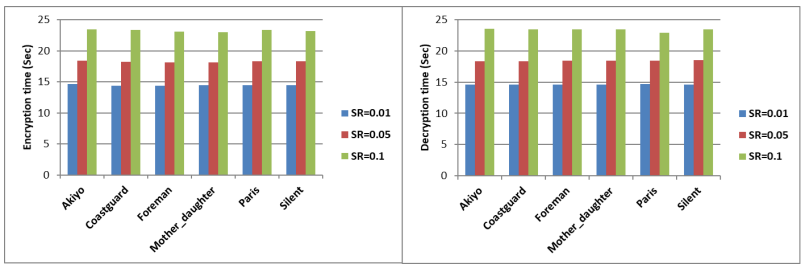
Method	Correlation Coefficient			NPCR	UACI	Entropy	Key space
	Horizontal	Vertical	Diagonal				
Unde *et al.* [34]	0.0030	0.0027	0.0014	99.69	33.55	7.9900	-
Karmakar *et al.* [2]	-0.0010	0.0057	0.0055	99.61	33.48	7.9987	-
Sethi *et al.* [58]	0.0020	0.0048	0.0114	99.62	33.48	**7.9993**	2^{264}
Song *et al.* [85]	-0.0245	0.0135	-0.0143	-	-	7.8708	10^{90}
Muhammad *et al.* [89]	0.0031	0.0026	0.0006	99.61	33.44	7.9900	2^{299}
El-Latif *et al.* [87]	-0.0012	0.0004	-0.0003	99.62	-	7.9984	2^{305}
Sethi *et al.* [4]	-0.0005	-0.0012	**-0.0001**	99.64	33.57	7.9989	$\mathbf{2^{512}}$
Hu *et al.* [119]	0.0030	0.0034	0.0098	99.59	33.51	7.9963	2^{200}
Meng *et al.* [110]	0.0076	−0.0051	0.0139	99.61	33.36	7.9975	2^{243}
K.U. *et al.* [109]	0.0005	0.0010	0.0005	99.61	33.46	7.9989	2^{384}
Chen *et al.* [108]	0.0046	0.0021	0.0035	99.63	33.47	7.9976	10^{135}
Our (Single Keyframe)	0.0007	-0.0008	0.0019	99.64	33.48	7.9989	10^{56}
Our (Double Keyframe)	**0.0005**	**0.0002**	**-0.0001**	**99.71**	**33.59**	**7.9993**	10^{56}

- indicate data are not available

With certain medical imaging technologies, long-term patient monitoring produces large volumes of data daily [64]. In order to reduce redundancies and speed up the acquisition process, it is necessary to compress the data using fewer samples for efficient transmission and analysis. In these circumstances, our proposed method not only protects patient information but also compresses the data using fewer samples.



(a) Single keyframe



(b) Double keyframe

Figure 5.6: Encryption and decryption times for single keyframe and double keyframes method on the first two GOPs of six CIF video sequences at three different sampling ratios

5.4.3 Statistical analysis

5.4.3.1 Correlation coefficient (CC)

The correlation coefficient between two pixels (p,q) in a frame can be found using the following equation:

$$CC_{pq} = \frac{cov(p,q)}{\sqrt{D(p) \times D(q)}} \tag{5.6}$$

where $cov(p,q)$ is the covariance between p and q. $D(p)$ and $D(q)$ are the variances of p and, q respectively.

Fig. 5.7 and Fig. 5.8 show the distribution of neighboring pixels in all directions for the Y component of Akiyo video for the original and encrypted GOP's keyframe and non-keyframes respectively. The encrypted keyframes and non-keyframes are uncorrelated. To defend against statistical attack, It is required that adjacent pixels in all directions be decorrelated to a high degree. Our method yields lesser correlation coefficients for video encryption. Consequently, our scheme is resistant to statistical attacks.

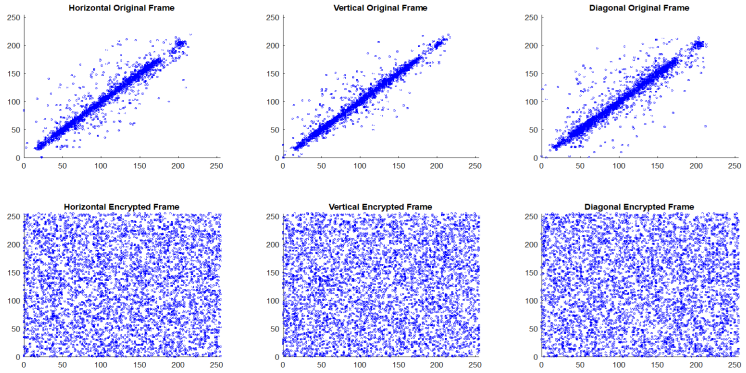


Figure 5.7: Correlation distribution of Akiyo video keyframe (first GOP) in Horizontal, Vertical, and Diagonal directions for SR = 0.1

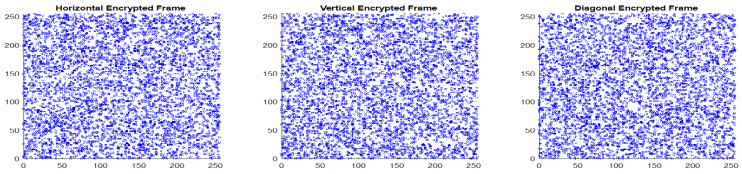


Figure 5.8: Correlation distribution of Akiyo video non-keyframe (first GOP) in Horizontal, Vertical, and Diagonal directions for SR = 0.1

5.4.3.2 Entropy and Histogram Analysis

Entropy is a statistical measure of uncertainty. In general, higher entropy represents more uncertainty. Encrypted video frames have an entropy value closer to 8, indicating that they are quite unpredictable. A video frame's entropy may be calculated as:

$$H(m) = -\sum_{j=1}^{255} P(m_j) \log_2 P(m_j) \tag{5.7}$$

where H(m) is the entropy, $P(m_j)$ indicates the probability of symbol m_j. In Fig. 5.9, we show the encrypted keyframe and non-keyframes of the Akiyo video for the first GOP for various sampling ratios. Histograms show the distribution of pixel intensity in video frames. A uniform histogram of encrypted video frames improves security. Fig. 5.10 shows a histogram of encrypted frames of Akiyo video for different sampling ratios. The histogram of encrypted frames is uniform. So, our technique can withstand statistical attacks.

5.4.4 Key Sensitivity and Avalanche Effect

We compute the difference between video frames using the number of bit change rate (NBCR) to quantitatively test the plaintext sensitivity. The NBCR between

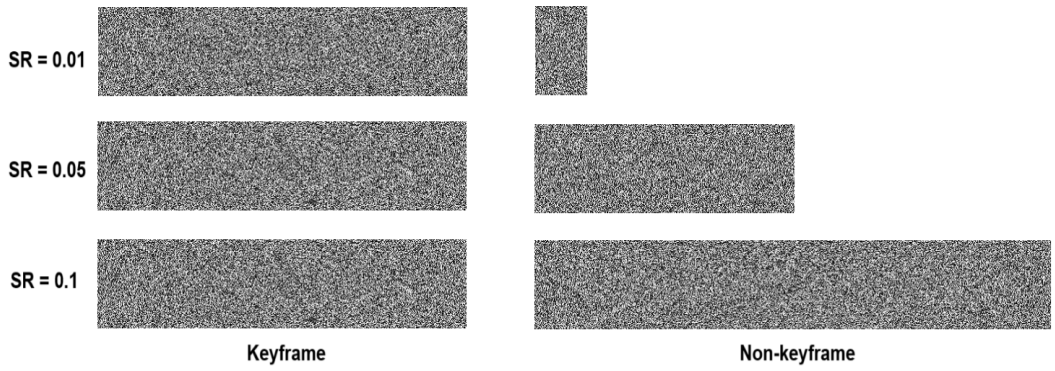


Figure 5.9: Encrypted frame of Akiyo video (first GOP) for different sampling ratios

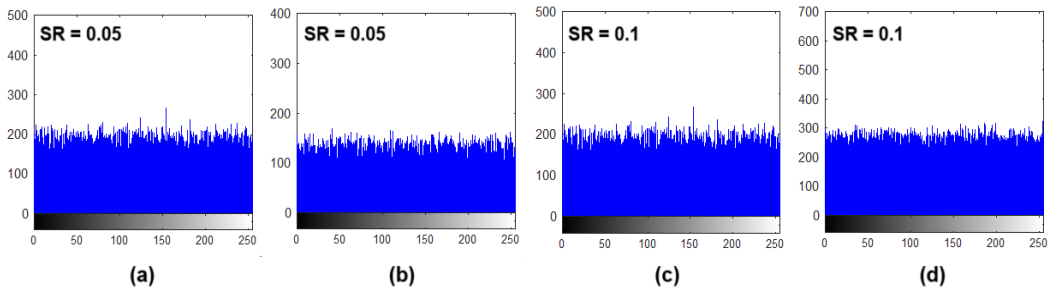


Figure 5.10: Histogram of Akiyo video (a, c) Encrypted keyframe and (b, d) Encrypted non-keyframe (first GOP) at different sampling ratios

two sequences S_1 and S_2 of length L can be expressed using Eq.(5.7), where HD is the Hamming distance. NBCR will get close to 50% if S_1 and S_2 are two statistically independent data sequences. When we change a plaintext by one bit for our proposed scheme, we get 50% NBCR.

$$NBCR = \frac{HD(S_1, S_2)}{L} \tag{27}$$

Fig.5.11 shows the encrypted versions of the Akiyo video keyframe with only one-bit difference in the plaintext. The two obtained encrypted frames are completely different, as both encrypted frames have a uniform distribution. Also, we show the difference between these two encrypted frames and found that the difference result gives uniform distribution. Our approach is more resistant to differential attacks as a result of the fact that decryption is impossible even if there is just a very little variation in the secret key. A little variation in the secret key will result in a decrypted frame that is different from the original frame, which is shown in Fig.5.12.

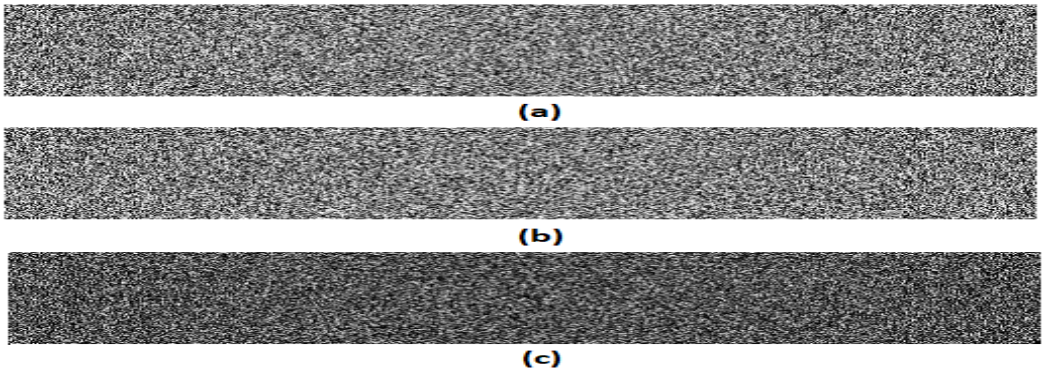


Figure 5.11: (a) Encrypted Akiyo video keyframe (b) Encrypted Akiyo video keyframe by changing a 1 bit in the plaintext (c) Difference between a and b

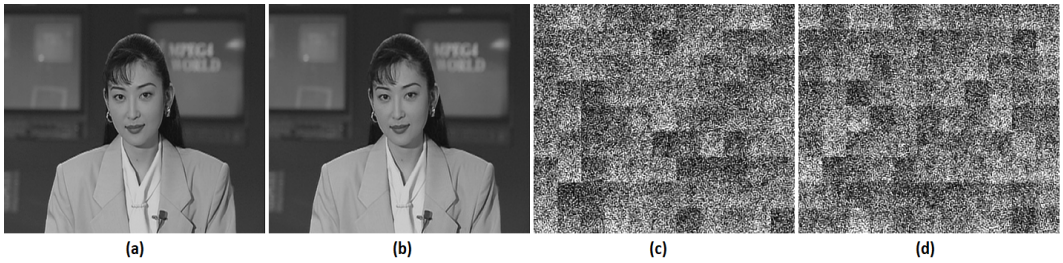


Figure 5.12: (a) Original keyframe of Akiyo video (b) decrypted keyframe with exact key $x_0 = 0.765$, $a_0 = 50$, $x_1 = 0.856$, $a_1 = 100$ (c) decrypted keyframe with key $x_0 = 0.765$, $a_0 = 50 + 10^{-14}$, $x_1 = 0.856$, $a_1 = 100$ (d) decrypted keyframe with $x_0 = 0.765 + 10^{-14}$, $a_0 = 50$, $x_1 = 0.856$, $a_1 = 100$

5.4.5 Robustness Analysis

Network failure and congestion are typical problems during data transmission [120]. It is possible that a portion or the whole of the encrypted video frames may be lost during transmission. Therefore, it is crucial to ensure the robustness of the proposed video compression-encryption technique. Encrypted video keyframe and non-keyframes for the single keyframe-based model are cropped partially from the middle and left, and their corresponding recovered versions are shown in Fig.5.13 and Fig.5.14 respectively. From the figure, the information about the original video frames is recognizable with exact keys from the recovered frames. The quality of decrypted frames is acceptable if the encrypted frames are partially excluded, which indicates original video frame information diffused in the encryption process. So the proposed encryption scheme can defend against cropping attacks [121]. Also, we tested the robustness of the proposed method by adding different levels of Salt and Pepper (SP) noise to the encrypted frame. The PSNR and SSIM of the decrypted frame is increasing if the level of noise reduces. In Fig.5.15, we show the decrypted

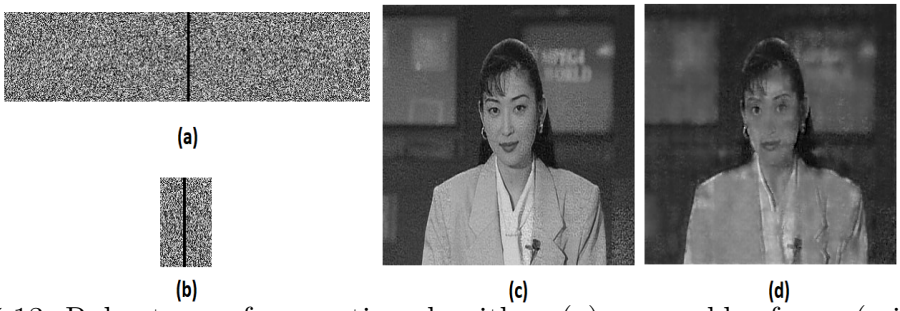


Figure 5.13: Robustness of encryption algorithm, (a) cropped keyframe (middle), (b) cropped non-keyframe (middle), (c) recovered keyframe, (d) recovered non-keyframe

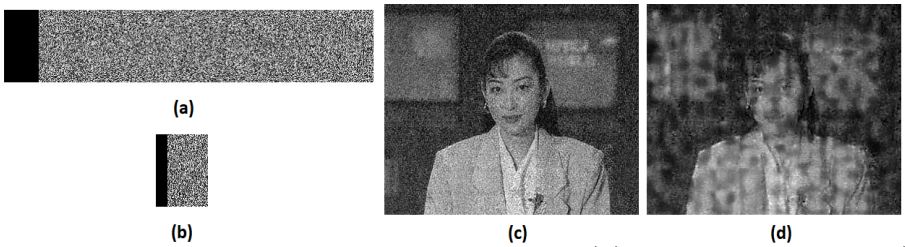


Figure 5.14: Robustness of encryption algorithm, (a) cropped keyframe (left), (b) cropped non-keyframe (left), (c) recovered keyframe, (d) recovered non-keyframe

Figure 5.15: Decryption of Foreman video frame for various degrees of Salt and Pepper noise

Foreman video frame for SP noise levels 0.0001, 0.001, and 0.01. The decrypted video frame is visibly distinguishable in the presence of noise which indicates our proposed scheme can defend against noise attacks.

5.5 Disscussion

In this work, we present a secure and robust video compression and encryption scheme with high-quality recovery. The proposed scheme applies deep compressive sensing-based video compression and DSDX-based video encryption. Use of multilevel feature compensation from the neighboring non-keyframes, along with the keyframes, in deep compressive sensing improves the average quality of the recovered video frames, even at lower sampling ratios. We compared our method with SOTA deep-learning-based image and video CS methods and found that our solution

performs better in terms of PSNR and SSIM of the recovered frames. Further, our proposed encryption model yielded better results with respect to mean CC, NPCR, UACI, and entropy compared to SOTA image and video encryption approaches. In the future, we plan to work on adaptive CS [122] which will be able to effectively and adaptively sample the video data.

Chapter 6

A Unified Video Compression and Encryption Scheme for WVSNs

In resource-constrained Wireless Visual Sensor Networks (WVSNs) with limited battery power, low bandwidth, and restricted node processing capacity, real-time secure video capture and transmission becomes an extremely challenging task. We propose a unified scheme consisting of adaptive parallel compressive sensing (APCS) based video compression, and a highly secure video encryption to achieve the above goal. At the heart of our solution lies a 1D chaotic sequence based on Log–Logistic–Cubic (LLC) map, known for its high unpredictability. We first detect shot boundaries of a video, following which APCS is applied to the frames within each shot. The LLC map is used to initialize measurement matrices to sample the frames. These matrices are further efficiently updated through QR decomposition via Givens rotations to enhance the quality of the recovered video. The suggested LLC map-based chaotic sequences are used both in permutation and key-based operation selection substitution steps of the encryption process. Our encryption method offers a large key space, high NPCR, low correlation coefficients, and higher PSNR with less overhead. Extensive comparisons with state-of-the-art video compression and video encryption methods clearly demonstrate the efficacy of our solution.

6.1 Introduction

A wireless visual sensor network (WVSN) consists of several spatially distributed camera nodes that monitor important areas. WVSN nodes process, store, and transmit raw video frames in wireless mode. They have widespread applications in surveillance, object tracking, environmental monitoring, and smart homes [123].

WVSNs require efficient data collection strategies to effectively aggregate within limited resources [124]. By implementing data compression at each sensor node, the volume of data for transmission can be reduced [125–127]. However, all the data compression approaches presented for WVSNs mainly concentrate on spatial redundancy. In [128], the authors introduced a block-based compressive sensing (BCS) algorithm for WVSNs using an iterative, multi-hypothesis reconstruction technique at the decoder, which leads to longer decompression times. Further, they did not address the problem of secure encryption.

We present here a unified video compression and encryption scheme by taking into account both spatial and temporal redundancies. Our approach not only reduces the time to recover the video frames from compressed samples with high-quality, but also protects the data from unauthorized access within the network. We employ discrete cosine transform (DCT) for sparsification of video frames and apply adaptive PCS for sampling frames in each shot to enhance both compression performance and quality of video recovery. A Log–Logistic–Cubic (LLC) map is used to initialize measurement matrix. This matrix is further updated by Givens rotations capitalizing on sparsity to yield high quality recovered video in real-time. For security, we propose a swap-based permutation and bi-directional substitution method utilizing LLC map-based chaotic sequences, ensuring small plaintext changes drastically alter ciphertext, enhancing defense against statistical and differential attacks. The proposed unified compression and encryption scheme at the shot level using adaptive PCS will be well suited for WVSNs, which require low power, bandwidth, and memory.

There are several key differences between our previously published paper [4] and the current work. The method [4] used fixed sampling ratios for all non-reference frames, which does not recover high-quality video. But, in our proposed scheme, we use APCS to get high-quality video after recovery. The encryption scheme used in [4] takes two rounds of operation to qualify all the encryption parameters. However, our proposed scheme takes only one round of operation, thereby enhancing the speed of operation. In [4], we suggested a combined compression and encryption system employing PCS across fixed-length GOPs, but found a discernible quality loss as the GOP size was increased, making them unsuitable for WVSNs. The contributions of the present work are now summarized below:

1. We introduce a new chaotic map that exhibits a higher Lyapunov exponent, greater sample entropy, and enhanced sensitivity to minor variations. Furthermore, it passes both the NIST and 0-1 randomness test, making it suitable for

video compression during sampling. Additionally, it is employed within both permutation and substitution techniques to enhance security measures.

2. We propose an APCS based scheme for compressing a video at the shot level. The proposed chaotic map is employed to generate the measurement matrix, which is then refined through QR decomposition utilizing Givens rotations to enhance recovery quality.

3. We propose an encryption scheme for securing compressed data using swap-based permutation and bidirectional substitution with modular addition, XOR, and circular shifting. Security is enhanced by selecting substitution operations from a chaotic map-based sequence. This framework is suitable for real-time and resource-constrained WVSNs.

The rest of the paper is organized as follows: In Section II, we discuss related works and highlight their limitations. In Section III, we describe our proposed unified video compression and encryption scheme in detail. Section IV presents the experimental results with analysis. Finally, the paper is concluded in Section V.

6.2 Related Work

6.2.1 Video compression using Compressive Sensing

The majority of data compression schemes use transform-based methods, non-transform methods, and distributed coding techniques during the acquisition of the data [129, 130]. Compared to conventional compression algorithms [131], [132], CS provides lower encoder complexity, making it a particularly useful method for data acquisition in sensor networks [133]. Block Compressed Sensing with Smooth Projected Landweber (BCS_SPL) is an adaptive image compression technique for WVSNs that was presented by Zhang et al. [134]. Each image block is given a fixed and an adaptive sampling rate in order to account for differences in statistical characteristics and reduce data redundancy. While the adaptive rate is adjusted based on the standard deviation of the relevant block, the fixed rate is established empirically to provide a minimal degree of reconstruction quality. However, combining fixed and adaptive sampling ratios increases transmission cost and computing complexity. Most data compression schemes in WVSN primarily focus on optimizing image coding at the sensor node, whether in a centralized or distributed configuration. In practical applications, sensor nodes often capture continuous video, either

at constant frame rates or triggered by motion detection, leading to the presence of both spatial and temporal redundancies. Although H.264 and MPEG 4 provide impressive compression ratios, Compressive Sensing (CS) presents a more appealing option for WVSNs due to its simpler encoder and significantly reduced storage requirements [135].

Block Compressive Sensing (BCS) methods [136] utilize the spatio-temporal correlations observed in images captured by sensor nodes. However, [136] offers poor reconstruction quality at low sampling ratios. Mun et al. [137] incorporated motion estimation and compensation (ME/MC) into BCS_SPL, resulting in the development of the MC_BCS_SPL video compression technique. However, compression efficiency and reconstruction quality are moderate for this scheme. Additional BCS-based methods [92, 95] employ multihypothesis (MH) based reconstruction to enhance the quality further. Multihypothesis is an iterative reconstruction process that offers good quality recovery, but requires a high recovery time, which may not be suitable for resource-constrained WVSNs. The author of [128] presented a block-based compressive sensing (BCS) video compression algorithm that outperforms current CS-based techniques in terms of compression ratio. The scheme is well-suited for WVSNs because it uses iterative, multihypothesis-based reconstruction at the decoder and second-stage compression at the encoder to achieve high compression ratios and better reconstruction quality without appreciably increasing encoder or decoder complexity. This method used an iterative process for the prediction of multiple hypotheses in the decoder, resulting in an increased decompression time. However, our scheme shows good results in terms of PSNR and SSIM values of recovered video frames compared to schemes used in [128]. We proposed a unified compression and encryption scheme for recovering high-quality video frames that incorporates adaptive parallel compressive sensing and a new encryption scheme.

6.2.2 Video Encryption

In recent years, different chaotic-map-based encryption schemes [2, 4, 9, 58, 118, 138–140] have been proposed to secure multimedia content. Teng et al. [139] introduced a method for encrypting multiple images employing an innovative spatiotemporal chaotic system, defined by dual dynamic coupling coefficients and random delayed feedback, which demonstrates outstanding security and robustness. However, the complexity of their coupled map lattice could hinder performance in real-time applications. The adaptive enhanced approximate message passing BCS (AE-AMP-BCS) technique, which integrates the established adaptive anti-aliasing filtering mecha-

nism to dynamically minimize noise effects during reconstruction, was used by the author to provide an image encryption algorithm based on BCS in [141]. Moreover, a chaotic system-based measuring technique and a twofold scrambling operation are used for image encryption. Its application in real-time or resource-constrained contexts may be limited by its comparatively high computational complexity. To achieve a secure image compression and encryption scheme, Gong et al. [140] suggested a four-dimensional chaotic system with improved logical operations. Their method changes pixel positions by applying the new fractal curve, which has a more complicated fractal structure. This procedure breaks up the pixel correlation. Their suggested measurement matrix with a low spectral norm is then used to compress the image. This process improves the reconstruction performance over the random measurement matrix. They used a bidirectional diffusion mechanism with a higher diffusion strength that came at the expense of a more complex approach. Lai et al. [142] proposed a novel approach to encrypting medical images using a memristive hyperchaotic system. They used a novel pixel split approach and the compressive sensing methodology to quickly lower the correlation between pixels that are next to each other. But when it comes to using standard video pipeline hardware, it has difficulties.

6.2.3 Unified Video Compression and Encryption

A combined video compression and encryption strategy using compressive sensing will be ideal for handling WVSN challenges. Sethi et al. [4] proposed a GOP-level video compression and encryption method utilizing PCS and an improved chaotic map. The system ([4]) used two rounds of operation to meet security needs, providing secure video encryption with excellent recovery. However, this approach uses a fixed GOP size, and the SR for frames is not adaptive. An increase in the GOP size results in a decrease in recovery quality, as the sparsity of the signal is influenced by the difference between two consecutive frames. For cloud computing applications, the authors of [110] created the retrieval image compression and encryption (RICE) technology, which provides a well-balanced approach for encryption, compression, and retrieval. However, both methods lack shot-level adaptive CS for compressing the data. In this paper, we introduce shot-level joint video compression along with adaptive PCS. Additionally, it incorporates an encryption scheme that features bidirectional substitution and key-based operations, which aim to enhance compression efficiency while providing improved security for wireless video sensor networks (WVSNs).

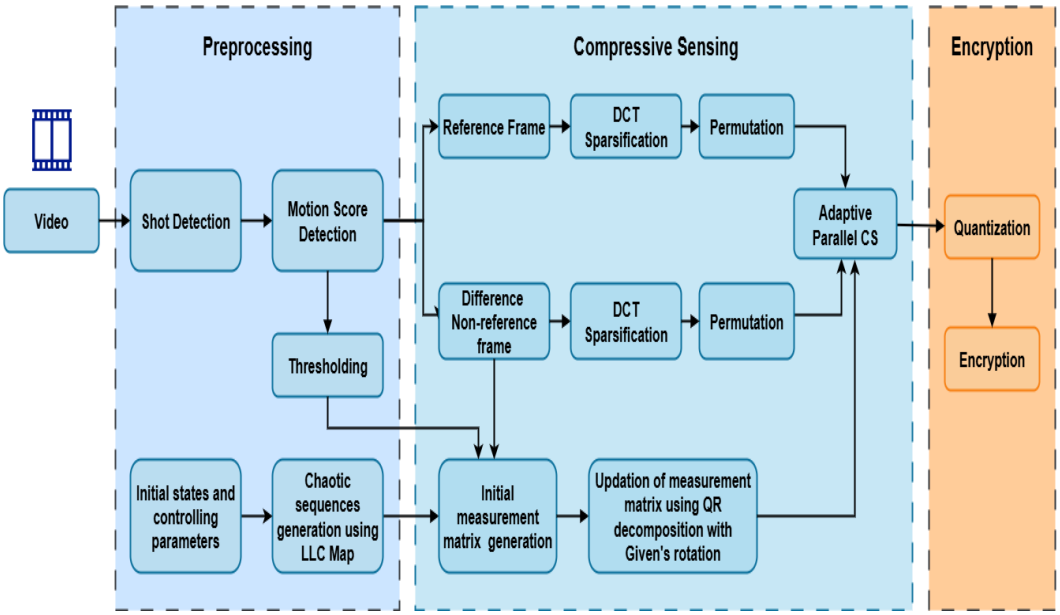


Figure 6.1: Block diagram of proposed unified video compression and encryption scheme

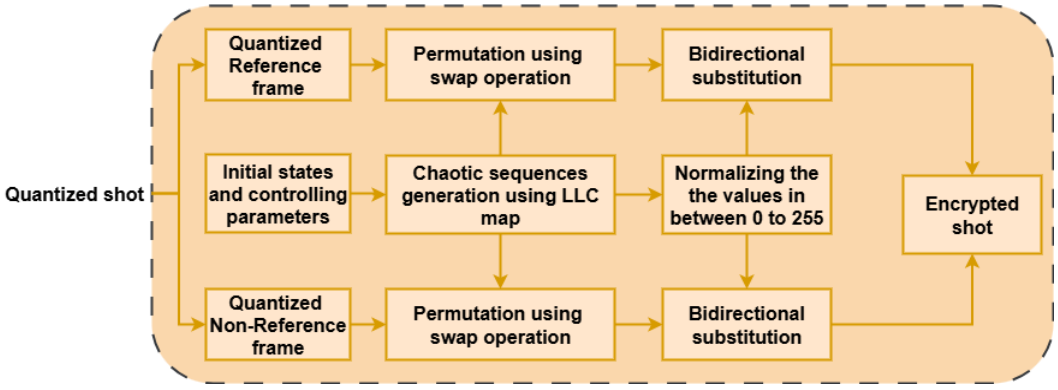


Figure 6.2: Block diagram of proposed encryption scheme

6.3 Proposed Method

This section uses six subsections to provide a detailed discussion of our suggested approach. In the first subsection, we outline the procedures of our proposed scheme. In the second subsection, we demonstrate the detection of the shot boundary in a video. In the third subsection, we discuss how to use adaptive parallel compressive sensing theory in our approach. We describe how chaotic sequences are generated and analyzed in the fourth subsection. The generation of the measurement matrix using the suggested chaotic map and QR decomposition with the Givens rotations is next discussed. In the last subsection, we explain our proposed encryption scheme. Fig. 6.1 shows the whole architecture of the suggested unified video compression and

encryption scheme. Fig. 6.2 illustrates the details of the encryption process, which only accepts quantized compressed video frames as input.

6.3.1 Shot Boundary Detection

Shot boundary detection divides a video into a number of shots. In this work, we detect the shot boundaries using the sum-squared difference (SSD) method [57], which calculate the total sum-squared difference of pixel intensities between two consecutive frames. The SSD method has low computational complexity, requires less memory, and does not require any separate training, thus making it suitable for the present problem of dealing with resource-constrained WVSNs. A threshold is applied on the already computed SSD values to obtain the shot boundaries.

6.3.2 Adaptive Parallel Compressive Video Sensing

We now propose a motion score ($MS_j^{(i)}$) for each frame j (of dimension $N \times N$) within a shot i, which is expressed as follows:

$$\mathrm{MS}_j^{(i)} = \sum_{m=1}^{N} \sum_{n=1}^{N} |\big(F_j^{(i)}(m,n) - F_{j-1}^{(i)}(m,n)\big)|, \tag{6.1}$$

where, $F_j^{(i)}(m,n)$ denotes the intensity of its constituent pixel at position (m,n), and likewise. Three sampling ratios are used as a part of making our solution pipeline adaptive. The first frame of every detected shot is designated as the reference frame and is sampled at a *high* sampling ratio. The sampling ratios for non-reference frames are determined based on their motion score values. The frames with significant motion are allocated a *medium* sampling ratio to retain the details, whereas, frames with less motion are sampled with a *low* sampling ratio to save bandwidth and energy. This adaptability is especially advantageous for WVSNs due to their restricted transmission capacity and battery life.

To apply compressive sensing, data must be represented in a sparse form. For our scheme, reference frames are sparsified using 2D DCT. We apply 2D DCT on the difference frames, obtained from every pair of consecutive non-reference frames of a given shot, for further sparsification. DCT coefficients for both reference and difference frames are permuted in descending order by absolute value, ensuring uniform sparsity across columns thereby increasing the PSNR of the recovered data [4]. We apply parallel compressive sensing (PCS), which enables efficient sampling and reconstruction by processing each column independently in parallel. The compressed

measurements of each column signal x_i are given by [143]:

$$\mathbf{y}_i = \mathbf{\Phi} \times \mathbf{x}_i = \mathbf{\Phi} \times \mathbf{\Psi} \times \mathbf{s}_i = \mathbf{\Theta} \times \mathbf{s}_i, \tag{6.2}$$

where, $\mathbf{\Phi}$ is the measurement matrix of size $M \times N$, $\mathbf{\Psi}$ is the orthonormal basis matrix of size $N \times N$, $\mathbf{\Theta}$ is the sensing matrix of the same size as $\mathbf{\Psi}$, and $\mathbf{y}_i$ is a M-dimensional measurement vector. So, there are altogether N number of $\mathbf{y}_i$'s in a frame, where N denotes the number of columns. The sampling ratio is defined as $SR = M/N$, where N = total number of samples in the original signal (or pixels in an image/frame) and M = number of compressive measurements acquired.

A measurement matrix is used to sample data in CS through a simple linear operation. The measurement matrix should be incoherent and orthogonal for high-quality data recovery. An initial measurement matrix is generated using the proposed LLC map, which has high mutual coherence and is non-orthogonal. Therefore, we update the initial measurement matrix using Givens rotations to minimize mutual coherence and make it orthogonal, ensuring high-quality data recovery.

6.3.3 Chaotic Sequence Generation

The chaotic range in a 1-dimensional chaotic map like Sine, Logistic, and Cubic is lower, making it more susceptible to brute-force attacks. Despite having a lower chaotic range, the 1-dimensional map has a lower complexity and is unpredictable within that lower chaotic range. Higher-dimensional chaotic maps, although offering a higher chaotic range and enhanced security, are substantially more complex. This complexity can lead to more computationally expensive operations, making them less efficient and challenging to implement in real-time resource-constrained systems.

The logistic map and cubic chaotic map are commonly used 1D chaotic maps. Mathematically, they are represented as:
Logistic map:

$$x_{i+1} = ax_i(1 - x_i) \tag{6.3}$$

Cubic map:

$$x_{i+1} = x_i^3 - bx_i \tag{6.4}$$

where a and b are the controlling parameter and $a \in [0, 1]$ for logistic, and $b \in [2.3, 3]$ for cubic map.

Many of the chaotic maps that are now in use only exhibit chaotic behavior within narrow parameter ranges; when their parameters are increased to high numbers, they are unable to depict chaos. This is because their phase planes will become unbounded as their parameters increase. In [144], a mod-based chaotification method, using modular operation as a constrained transformation to augment the chaotic complexity of 2D maps by expanding the chaotic range and unpredictability. We aim to achieve a larger chaotic range at a lower dimension with a lower level of complexity and a higher degree of unpredictability. So, we introduce a novel 1D chaotic map that combines logistic and cubic maps. Mathematically, our proposed chaotic map is expressed as:

$$x_{i+1} = \text{mod}\left(\frac{1000\,|\log(ax_i(1-x_i))|}{|\log(x_i^3 - bx_i)|}, 10\right) \tag{6.5}$$

We call the proposed chaotic map the Log-Logistic-Cubic (LLC) map.

6.3.3.1 Analysis of Proposed Chaotic Map

We select measures like bifurcation diagram, lyapunov exponent, and sample entropy to evaluate our proposed chaotic map. We have also tested the sensitivity to initial conditions and controlling parameters for the proposed map by applying a minor change in both parameters.

6.3.3.2 Fixed Point and Stability

A dynamical system's fixed point is a crucial feature that may reveal the dynamic nature of the system. Here, we determine the fixed points of the proposed map and examine their stability.

The fixed points of the proposed map are represented by x^*. A point x^* is called a fixed point of the map if it satisfies

$$x^* = f(x^*) = \text{mod}\left(\frac{1000\,|\log(ax^*(1-x^*))|}{|\log((x^*)^3 - bx^*)|}, 10\right). \tag{6.6}$$

To find the fixed points, let an auxiliary variable be defined as

$$y^* = \frac{1000\,|\ln(ax^*(1-x^*))|}{|\ln((x^*)^3 - bx^*)|}. \tag{6.7}$$

Then, the fixed-point condition becomes

$$x^* = \mathrm{mod}(y^*, 10). \tag{6.8}$$

Unwrapping the modulo operation yields

$$y^* = x^* + 10k, \qquad k \in \mathbb{Z}. \tag{6.9}$$

so that substitution gives the branchwise equation

$$1000\,|\log(ax^*(1-x^*))| = \left|\log\left((x^*)^3 - bx^*\right)\right|(x^* + 10k), \qquad k \in \mathbb{Z}. \tag{6.10}$$

Finally, by imposing the domain constraints

$$x^* \in [0, 10), \qquad ax^*(1-x^*) > 0, \qquad (x^*)^3 - bx^* > 0, \tag{6.11}$$

which ensure that both logarithmic terms are well defined, we can solve the above equation for each integer k to obtain all fixed points of the map.

The fixed point of a chaotic system might be in either a stable or unstable condition. The stability of the system is related to the slope of its curve at that point. The system's gradient (J) at a fixed point determines its stability. The fixed point achieves superstability at $J = 0$, stays stable at $0 < |J| < 1$, becomes neutral at $|J| = 1$, and becomes unstable at $|J| > 1$, which causes neighboring attractors to diverge. Please see Appendix for the Jacobian (J) of the proposed LLC chaotic map.

6.3.3.3 Bifurcation Diagram

It is possible for the bifurcation diagram of a dynamical system to take up some areas of its phase space. When the degree of density in the bifurcation diagram is lower, it suggests that the chaotic signals of the associated dynamical system are more uniform. Therefore, the bifurcation diagram can provide a visual representation of the dynamics of a dynamical system.

For good chaotic performance, the attractor has to occupy the maximum regions in phase space. We compare the results with two base maps i.e, logistic and cubic. We consider $a = 1000, b = 50$ as controlling parameters and $x = 0.567$ as the initial state, and iterate 2000 times for every map. Next, we plot all the points in phase space. Both logistic and cubic maps exhibit chaotic behaviour within restricted parameter ranges, and their bifurcation diagrams in Fig. 6.3 clearly show that their

outputs do not spread out uniformly. In Fig. 6.3, we have also shown the bifurcation diagram of the proposed map in the parameter space (a, b) where $a \in [0, 1000]$ and $b \in [0, 50]$. From Fig. 6.3, we can see that the generated map is distributed in maximum regions of phase space within $(0, a)$ and $(0, b)$, which indicates the suggested map is highly unpredictable and has good ergodicity. The suggested map shows complex, chaotic behavior in all the parameter values. The complete output range corresponds to the uniformly distributed output values. Because of its output's uniform distribution, the suggested map may be used to generate pseudorandom sequences.

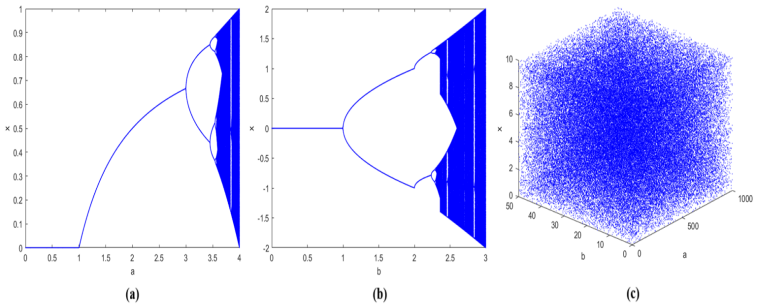


Figure 6.3: Bifurcation diagram (a) logistic map, (b) cubic map, (c) LLC map in parameter (a,b) space

6.3.3.4 Lyapunov exponent(LE)

It is common practice to test chaos of any dynamical system with the LE [145]. Through the use of deterministic equations, it is able to test chaotic dynamical systems. In a certain amount of unit time, the LE calculates an estimate of the average separation rate between two dynamical system trajectories that are extremely close to the initial states. When the lyapunov exponent of a dynamical system is positive, it indicates that even minor changes in the system's initial conditions may significantly impact the outcomes that are produced. Because of this, the presence of a single positive LE points to chaotic dynamics, as long as other conditions are met, like the phase space being compact [146]. A higher positive LE suggests that the rate of divergence is occurring at a faster pace.

The Lyapunov Exponent for equation $x_{i+1} = G(x_i)$ is mathematically defined as follows:

$$\lambda_{G(x)} = \lim_{n \to \infty} (1/n) \ln \mid (G^n(x_0 + \epsilon) - G^n(x_0))/\epsilon \mid \tag{6.12}$$

where value of ϵ is very small.

The logistic and cubic maps have better chaotic behaviour in a limited range and low LE. In Fig. 6.4, we have shown the LE of both logistic and cubic maps. The proposed chaotic map encompasses the entire range of controlling parameters for which LE is positive. Fig. 6.5 illustrates the distribution of LE for the controlling parameter (a) within the interval (0,1000) and b within the interval (0,50).

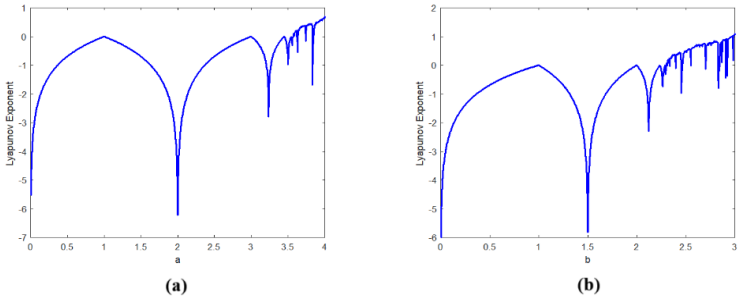


Figure 6.4: Lyapunov exponent of (a) logistic map and (b) cubic map

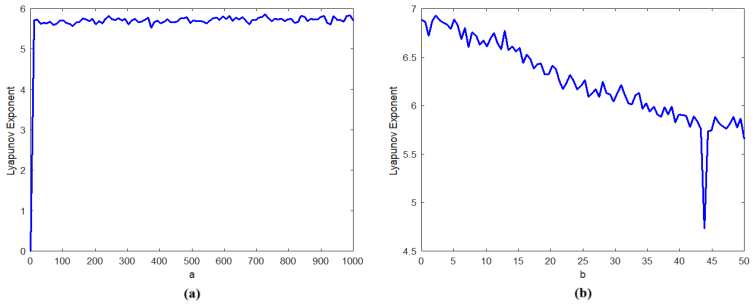


Figure 6.5: Lyapunov exponent of proposed chaotic map for (a) parameter a and (b) parameter b

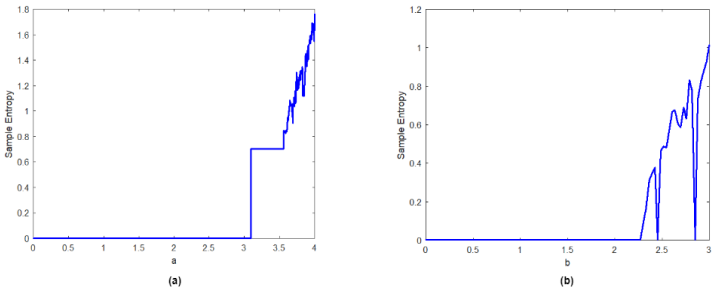


Figure 6.6: SE of (a) logistic map and (b) cubic map

6.3.3.5 Sample Entropy (SE)

The sample entropy may be used to quantify a time series' complexity [147]. Reduced regularity in the time series is indicated by a greater positive SE. A larger positive

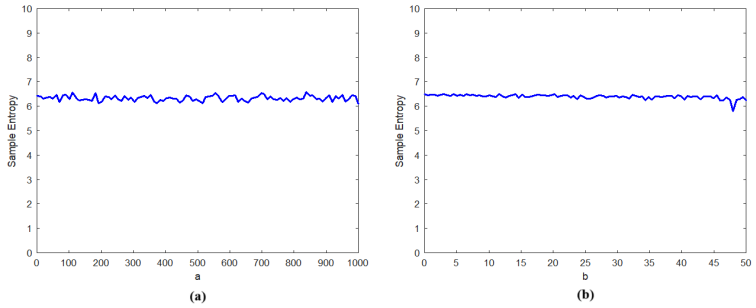


Figure 6.7: SE of proposed chaotic map for (a) parameter a and (b) parameter b

Table 6.1: CC of chaotic sequences generated from the proposed chaotic map

$CC(S_1, S_2)$	$CC(S_1, S_3)$	$CC(S_1, S_4)$
0.0004	-0.0079	0.0076

SE indicates more complexity when the observed time series is produced by a chaotic system. In our experiments to compute the SE of proposed chaotic map, we set the parameter m to 2, and r was defined as 0.2 times the standard deviation of the time series, following the methodology outlined in [147]. In Fig. 6.6, we have shown the SE of the logistic map and cubic map. We have shown the SE of time series produced by the suggested chaotic map in Fig. 6.7. The SE of the time series generated from our proposed map is higher than the SEs of the time series generated from existing maps.

6.3.3.6 Sensitivity Analysis of Chaotic Sequence

Sensitivity to the initial state is an important property of the chaotic map. The proposed chaotic map is very sensitive to initial state and controlling parameters.

We generate four chaotic sequences S_1, S_2, S_3 and S_4 of length 50000 by 1D proposed map considering a minor change in control parameters and initial state. The initial state and controlling parameters for S_1, S_2, S_3 and S_4 are (x_0, a, b) as $(0.567, 1000, 50), (0.567 + 10^{-7}, 1000, 50), (0.567, 1000 + 10^{-7}, 50)$ and $(0.567, 1000, 50 + 10^{-7})$ respectively. We observe that even for the minor change in the control parameter and initial state, the generated sequences look quite different from each other. Furthermore, we calculate the correlation coefficients $CC(S_1, S_2)$, $CC(S_1, S_3)$, and $CC(S_1, S_4)$ in Table 6.1 and find that CCs are approaching 0, which indicates that the proposed chaotic map is sensitive to the initial state and controlling parameters.

6.3.3.7 Randomness test of Chaotic Sequence

NIST: A well-known test standard for assessing the randomness of a number is the National Institute of Standards and Technology SP800-22 [148]. Each of its 15 subtests aims to evaluate the non-randomness aspect of a series of random numbers from various perspectives. The random number sequences may pass a subtest if the corresponding p-value exceeds the experimental significance level α. There should be more sequences than α^{-1} in a single test, and the test binary sequence consists of one million bits. According to [148], we use one million bits as the testing input produced by our suggested chaotic map, with alpha = 0.01. We have verified that the p-value for all subtests exceeds 0.01, indicating that each subtest successfully passes the randomness test and that the chaotic map exhibits high randomness.

6.3.3.8 0-1 Test

0-1 test is used to know the chaotic behavior of a random sequence [138]. The presence of chaos is assessed by examining if the output result approaches 1. If the result is 1, then the discrete sequence follows the Brownian motion. The mathematical expression for the 0-1 test is given as follows:

$$P(n) = \sum_{i=1}^{n} \Psi(i)\ \cos\big(ic\big), n = 1, 2, 3, \ldots \tag{6.13}$$

$$Q(n) = \sum_{i=1}^{n} \Psi(i)\ \sin\big(ic\big),\ \ n = 1, 2, 3, \ldots \tag{6.14}$$

where $\Psi(i)$ is the discrete sequence and c is a real random number. The system is in a state of chaos when the (P,Q) plot shows scattered and unpredictable movements. The results of the 0-1 test performed on the proposed chaotic map with different parameter settings are shown in Fig. 6.8

6.3.4 Measurement Matrix Generation

Chaotic maps can be used to generate the measurement matrix (MM) for compressive sensing (CS). The use of chaotic maps can be advantageous because their intrinsic complexity and randomness can produce low coherence and ensure enhanced signal recovery. Initially, a measurement matrix is generated from the proposed LLC chaotic map using [4]. We then employ Givens rotations [149], instead of the tra-

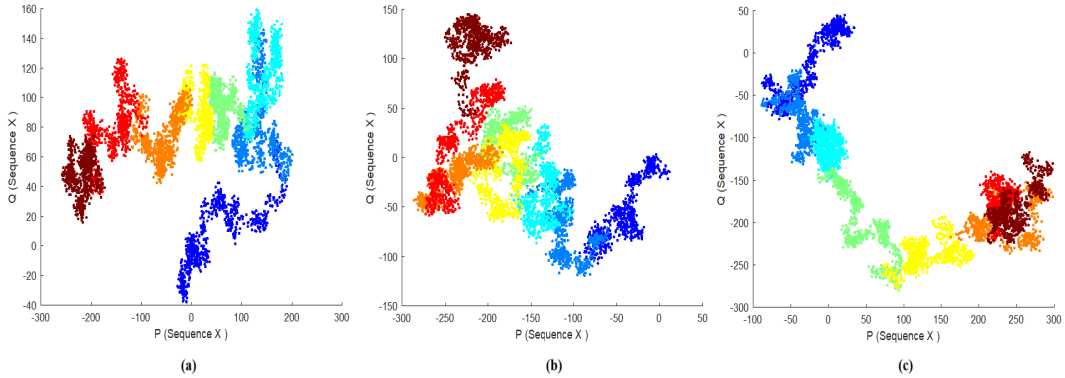


Figure 6.8: $0-1$ test of the proposed map. (a) Sequence X with $a_1 = 10, b_1 = 50, x_0 = 0.765$ (b) Sequence X with $a_1 = 25, b_1 = 15, x_0 = 0.385$ (c) Sequence X with $a_1 = 400, b_1 = 100, x_0 = 0.567$.

ditional Householder transform, to update the initial measurement matrix through QR decomposition. Instead of reflecting a column vector as is done in a Householder Transformation [150], using Givens rotation, we rotate it to obtain an ensemble of orthogonal vectors. In contrast to Householder transformations that involve dense reflectors across full submatrices, Givens rotations focus on working locally with two rows at a time. Its localized strategy greatly reduces computational overhead, especially when dealing with nearly orthogonal matrices, such as, those produced by the initially generated measurement matrix using LLC map. Another benefit of Givens rotations over the Householder transformations is their straightforward parallelization capability, making them well-suited for our column-by-column data processing approach. Each Givens rotation requires merely a handful of trigonometric and multiplication operations, making it ideal for WVSN nodes.

Let $\mathbf{\Phi}_1 \in \mathbb{R}^{m \times n}$ be the initial measurement matrix generated from the LLC map. To transform it into an orthogonal measurement matrix, QR decomposition is employed through Givens rotations. A Givens rotation $\mathbf{G}_{p,q}(\theta)$ is an orthogonal matrix specifically designed to modify only the p^{th} and q^{th} rows, aiming to zero-out chosen element $R_{q,j}$ below the main diagonal as

$$\mathbf{G}_{p,q}(\theta) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & \cdots & -s & \\ & & \vdots & \ddots & \vdots & \\ & & s & \cdots & c & \\ & & & & & \ddots \end{bmatrix}, \quad c = \frac{a}{\sqrt{a^2+b^2}}, \quad s = -\frac{b}{\sqrt{a^2+b^2}}, \tag{6.15}$$

where $a = R_{p,j}$ and $b = R_{q,j}$. Starting with $\mathbf{Q} = \mathbf{I_m}$ and $\mathbf{R} = \mathbf{\Phi_1}$, each rotation

updates as:

$$\mathbf{R} \leftarrow \mathbf{G}_{p,q}(\theta)\,\mathbf{R}, \quad \mathbf{Q} \leftarrow \mathbf{Q}\,\mathbf{G}_{p,q}(\theta)^T \tag{6.16}$$

Executing all necessary rotations to remove elements below the main diagonal is equivalent to

$$\mathbf{Q}^T\mathbf{R} = \left(\prod_k \mathbf{G}_{p_k,q_k}(\theta_k)\right)\mathbf{\Phi_1}, \tag{6.17}$$

where, Q is orthogonal and R is upper triangular. Finally, the refined measurement matrix is obtained as $\Phi_2 = Q$.

6.3.5 Encryption of compressed data

The quantized frames in each shot are permuted using a swap-based scheme where a new location is found by using the indices of two chaotic sequences generated from the proposed chaotic map. Two chaotic sequences are generated and sorted to find their indices. The new location for a sample in a quantized frame in a shot is found by considering the indices of the first and second sequences.

Let $P = [\,p_1, \ldots, p_N\,]$, $S^{(1)} = [\,s_1^{(1)}, \ldots, s_N^{(1)}\,]$, and
$S^{(2)} = [\,s_1^{(2)}, \ldots, s_N^{(2)}\,]$ where P is the quantized frame in 1D form in the shot and S^1 and S^2) are chaotic sequences generated from the proposed LLC chaotic map whose lengths are the same length of P. We can find the indices I_1 and I_2 as follows: $I_1 = sort(S^1)$ and $I_2 = sort(S^2)$ where I_1 and I_2 are indices of the sequence S^1 and S^2, respectively.

We construct a permutation. t: $(1, \ldots, N) \rightarrow (1, \ldots, N)$ by t(i) = j s.t. $I_2(j) = I_1(i)$, i=1,. . . ,N.

We can permute P by swapping operation $p_i \longleftrightarrow p_{t(i)}$

Since the histogram of the data before and after permutation remains the same, better security cannot be achieved by just permuting the data. Therefore, a substitution mechanism is necessary to modify the values considering the chaotic sequence, which can provide enhanced security for the permuted data. Using a pseudorandom sequence, the substitution method changes the pixel values so that the attacker cannot determine the original value. We employ modular addition, XOR, and circular shift in the substitution approach, which makes our scheme secure. We design a new bidirectional substitution scheme that considers both forward and backward substitutions, where the operations are selected using the proposed chaotic map.

We apply forward substitution to the permuted frame in a shot, as described in eq.6.18.

$$\begin{cases} C(i) = diff(P_1(i), op_1(i), diff(IV_1, op_2(i), S_1(i))) & \text{if } i = 1 \\ C(i) = C(i) >>> T_1 & \\ C(i) = diff(C(i-1), op_1(i), diff(P_1(i), op_2(i), S_1(i))), & \text{if } i \neq 1 \\ C(i) = C(i) >>> T_1 & \end{cases} \tag{6.18}$$

where the diff(x,y,z) function represents the execution of operation y between x and z. P_1, IV_1, T_1, $>>>$ and $<<<$ are the permuted quantized frame in the shot, the initial value, the shifting parameter, the left circular shift, and the right circular shift, respectively. op_1 and op_2 are the operations selected from the set of two operations, i.e., XOR and modular addition. S_1 and S_2 are generated from the proposed chaotic map. op_1 and op_2 are the sequences obtained by applying the mod 2 operation on $10^{10} \times S_1$ and $10^{10} \times S_2$, respectively.

We apply backward substitution to the forward-substituted data as described in eq.6.19.

$$\begin{cases} C1(i) = diff(C(i), op_1(i), diff(IV_2, op_2(i), S_2(i))) & \text{if } i = N \\ C1(i) = C1(i) <<< T_2 & \\ C1(i) = diff(C(i+1), op_1(i), diff(C(i), op_2(i), S_2(i))), & \text{if } i \neq N \\ C1(i) = C1(i) <<< T_2 & \end{cases} \tag{6.19}$$

where IV_2 is the initial value used in the backward substitution process, and T_2 represents the shift parameter.

6.4 Experimental Analysis

This section talks about evaluating the performance of the suggested video compression and encryption scheme using experimental analysis. To evaluate compression performance, we use peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). All experiments are performed on a desktop PC with an i5-9300H CPU having 2.40 GHz clock speed and 8 GB of RAM. We used MATLAB R2019b in the Windows 11 environment for our implementation. In the case of different videos and they are Susie (Sus), Coastguard (CG), Hall Monitor (HM),

Algorithm 11 A Unified Video Compression and Encryption Algorithm

Input: (Original video, initial values and controlling parameters, SRs)
Output: Compressed and Encrypted video

1: Divide the video into different shots using SSD
2: Divide each shot into reference and non-reference frames, and then further categorize the non-reference frames into groups with low and high motion frames
3: Generate a measurement matrix for the reference frame and different non-reference frames with different sizes as per the motion score
4: Apply adaptive PCS to each shot using generated measurement matrices
5: Encrypt each compressed shot using swap-based permutation and novel substitution mechanism ▷ (eq.14 and eq.15)
6: Assemble all compressed and encrypted shots

Mother-Daughter (MD), News (News), Container (Cont), Bus (Bus), and Football (FB). These datasets illustrate various scenarios with slow, moderate, and rapid movement features. Compressive sensing and reconstruction are performed for a total of 24 frames by considering the shot detection mechanism. We consider adaptive CS for compressing and encrypting the shots in a video.

6.4.1 Performance Analysis for Video Compression

6.4.1.1 Quantitative and qualitative validations with State-of-the-Art Methods

We compared our results with various video compression [92, 95, 128, 133, 136, 137]. For fair comparison, we used PSNR and SSIM for a group of 24 consecutive video frames from the standard test video datasets. We obtained the average PSNR and SSIM for the 24 video frames by treating them as various shots of various sizes, as shown in Table 6.2. We have obtained the best results in both PSNR and SSIM for all videos. We use the basis pursuit approach from the l_1-magic package [100] to recover the sparse video data. We show how the proposed video compression approach discussed in Section 3 is better than various other methods by comparing the average PSNR and SSIM of the recovered video and the time it takes to recover the video. In Table 6.2, we show a comparison of the average PSNR and SSIM of the recovered video with different methods. We compare the reconstruction time with many other schemes in Table 6.3. All other schemes require more time than ours. Compared to other methods, our suggested one has a lower reconstruction time and improved PSNR and SSIM. The best results are shown in bold in Tables 6.2 and

Table 6.2: Average PSNR and SSIM of recovered videos using different compressive sensing schemes

Videos	IFR [136]		MC_BCS_SPL [137]		MH-Tik [95]		MH-RTIK [92]		NLR-CS [133]		[128]		Our	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Susie	30.05	0.8	34.32	0.9	31.34	0.84	31.12	0.83	34.32	0.86	34.34	0.9	**36.33**	**0.9101**
CG	25.68	0.59	27.61	0.71	26.4	0.63	26.26	0.62	26.78	0.63	27.61	0.71	**29.17**	**0.7267**
HM	22.96	0.74	28.29	0.89	24.45	0.8	24.26	0.79	26.48	0.83	28.68	0.9	**30.74**	**0.9959**
MD	32.51	0.86	39.68	0.96	34.7	0.9	33.93	0.89	37.51	0.91	40.15	0.97	**42.3**	**0.9702**
News	24.35	0.78	31.89	0.94	26	0.84	25.6	0.82	31.1	0.9	32.41	0.95	**42.77**	**0.9775**
Cont	23.6	0.7	29.95	0.91	25.1	0.77	24.91	0.76	28.35	0.82	30.31	0.91	**33.8**	**0.9438**
Bus	22.19	0.59	22.52	0.63	22.46	0.6	21.9	0.6	22.5	0.62	22.52	0.64	**31.2**	**0.7801**
FB	21.9	0.57	25.78	0.71	23.57	0.6	22.21	0.6	24.5	0.68	25.1	0.69	**30.81**	**0.7715**

6.3.

In Fig. 6.9, we compare the visual quality of our method on various shot frames for the Hall monitor video sequence where the reference frame is sampled at SR = 0.7, the high-motion non-reference frame is sampled at SR = 0.3, and the low-motion non-reference frame is sampled at SR = 0.1. The recovered frames have better quality. The reference frame, high-motion non-reference frame, and low-motion non-reference frames have PSNRs of 31.89, 30.48, and 29.83, respectively.

Figure 6.9: Video quality comparison of our methods on various shot frames for the Hall monitor video sequence. (a) original frame (b) recovered reference frame (c) recovered non-reference frame with high motion and (d) recovered reference frame (c) recovered non-reference frame with low motion

Fig. 6.10 presents the PSNR values for the recovered frames of HM and MD videos. The PSNR of reference frames is marked with green dots, while other colors represent the PSNR for non-reference frames. The figure illustrates that, within a given shot, the difference in PSNR for non-reference frames is less than that observed

Table 6.3: Decompression time(sec.) for different methods

Video	IFR [136]	MC_BCS_ SPL [137]	MH-Tik [95]	MH-RTIK [92]	NLR-CS [133]	[128]	Our
Susie	1.57	14.8	2.41	2.41	41.2	17.09	**0.426**
CG	1.05	17.72	2.03	1.96	49.63	21.46	**0.651**
HM	1.65	16.03	3.11	2.5	41.58	25.69	**0.657**
MD	1.15	12.91	2.13	2.04	48.43	16	**0.575**
News	2.2	18.66	3.44	3.16	49.34	24.27	**0.618**
Cont	1.49	15.57	2.92	2.42	49.34	25.13	**0.905**
Bus	1.7	17.56	2.82	2.11	42.54	22.57	**0.435**
FB	1.9	19.35	3.05	2.34	44.37	20.73	**0.363**

Table 6.4: Average PSNR of different videos with two QR decomposition mechanisms

Video	Givens Rotation			Householder Reflection		
	PSNR	SSIM	Decomp time	PSNR	SSIM	Decomp time
Susie	36.33	0.9101	0.426	32.17	0.8208	0.8357
CG	29.17	0.7267	0.651	29.06	0.7164	0.774
HM	30.74	0.9959	0.657	30.53	0.7901	1.0256
MD	42.3	0.9702	0.451	42.1	0.9655	0.8192
News	42.77	0.9775	0.618	42.31	0.967	0.8604
Cont	33.8	0.9438	0.551	29.79	0.7298	0.8896
Bus	31.2	0.7801	0.435	31.05	0.78	0.7652
FB	30.81	0.7715	0.363	28.95	0.6907	0.7591

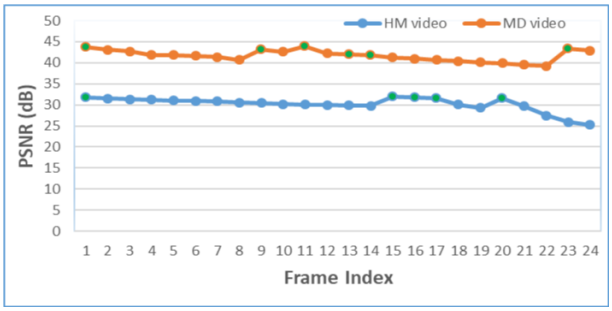


Figure 6.10: PSNR comparison for individual frames

for reference frames, resulting in an increased average PSNR for the recovered video. Non-reference frames maintain a nearly constant PSNR due to the selection of SRs based on motion scores. The minimal variation in PSNR between reference and non-reference frames within a shot, coupled with the absence of notable declines during

Table 6.5: Comparison of average CC, NPCR, UACI , entropy and key space of different schemes

Methods	CC			NPCR	UACI	Entropy	Key Space
	Horizontal	Vertical	Diagonal				
Sethi et al. [58]	0.0020	0.0048	0.0114	99.62	33.48	7.9993	2^{264}
Sethi et al. [4]	-0.0005	-0.0012	-0.0001	99.64	33.57	7.9989	2^{512}
Teng et al. [151]	-0.0016	0.0003	0.0006	99.59	33.41	7.9914	2^{512}
Li et al. [141]	-0.0015	0.0074	-0.0023	99.59	33.35	7.9944	-
Gong et al. [140]	-0.0015	-0.0072	0.0007	99.61	33.84	7.9987	2^{399}
Lai et al. [142]	-0.0005	-0.0010	-0.0008	99.60	33.46	7.9960	2^{256}
Karmakar et al. [2]	-0.0010	0.0057	0.0055	99.61	33.48	7.9987	-
Song et al. [85]	-0.0245	0.0135	-0.0143	-	-	7.8708	10^{90}
Muhammad et al. [89]	0.0031	0.0026	0.0006	99.61	33.44	7.99	2^{299}
Hu et al. [119]	0.0030	0.0034	0.0098	99.59	33.51	7.9963	2^{200}
Gan et al. [152]	-0.0029	0.0058	-0.0025	99.61	33.46	7.9986	2^{260}
Teng et al. [139]	0.0016	0.0028	-0.0010	99.61	33.45	7.9993	2^{400}
Zhang et al. [153]	0.0025	-0.0008	-0.0001	99.61	33.47	7.9980	2^{882}
Ding et al. [154]	-0.0003	-0.0005	-0.0010	99.61	33.46	7.9969	2^{425}
Our	-0.0005	-0.0008	- 0.0010	99.62	33.57	7.9952	2^{418}

- indicate data are not available

content transitions, illustrates that the adaptive approach effectively exploits temporal redundancy and ensures consistent quality, thereby increasing average PSNR for the video sequences.

6.4.1.2 Ablation Studies

We performed an ablation study, detailed in Table 6.4, on our updated measurement matrix by comparing two orthogonalization methods. Initially, the measurement matrix was created using our proposed chaotic map. Subsequently, it was modified using Givens rotations in one experiment and Householder reflections in another. The Givens rotation-based matrix consistently produced marginally better PSNR and SSIM scores compared to the Householder version and notably reduced the decompression time for each video. For dense matrices, the Householder-based transformation is appropriate, whereas Givens rotations are more suitable for handling sparse matrices [149].

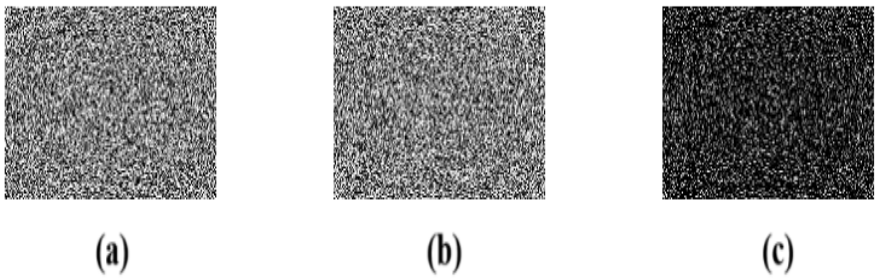


Figure 6.11: (a) Encrypted frame of Coastguard video (b) Encrypted frame of Coastguard video by changing a single bit (c) Difference between a and b

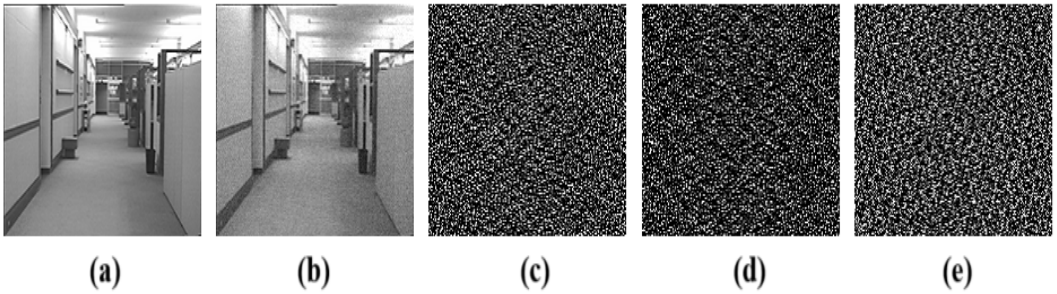


Figure 6.12: (a) Original frame Hall monitor video (b) decrypted frame with exact key $x_0 = 0.623$, $a_1 = 500$, $b_1 = 50$ (c) decrypted frame with key $x_0 = 0.623 + 10^{-14}$, $a_1 = 500$, $b_1 = 50$ (d) decrypted frame with $x_0 = 0.623$, $a_1 = 500 + 10^{-14}$, $b_1 = 50$ (e) decrypted frame with $x_0 = 0.623$, $a_1 = 500$, $b_1 = 50 + 10^{-14}$

6.4.2 Performance Analysis of Video Encryption

Here, we compare our encryption strategy with several state-of-the-art techniques [2, 4, 58, 85, 89, 119, 139, 140, 142, 151–154]. Table 6.5 shows a comparison of our proposed scheme with a number of recent video encryption schemes, considering mean CC, entropy, NPCR, UACI, and key space values for encrypted video. To prevent differential attack, the unified averaged changed intensity (UACI) and number of changing pixel rate (NPCR) values must be greater than the predicted ideal values of 33.4635 and 99.5693, respectively [118]. Our scheme produces NPCR and UACI values more than ideal values that can protect against differential attacks.

For security, encrypted frames must have low mean correlation coefficients. Our studies show that our encryption approach completely decorrelates neighboring pixels and is consequently very secure, since the mean CC for encrypted shot frames is closer to zero. We discovered that our proposed approach produces encrypted frames with a low mean correlation coefficient, indicating substantially uncorrelated data. Our method attains an entropy value approaching 8, indicating that the encrypted video frames exhibit a high degree of randomness. Both reference and non-reference

frames show uniform histograms, which helps our encryption scheme defend against statistical attacks. Our proposed method considers 9 parameters for generating the chaotic sequence. Each parameter has a precision of 10^{14}. So the total key space is $10^{14\times9} = 10^{126} \approx 2^{418}$. As a result, our method is not susceptible to brute-force attacks. Our scheme significantly reduces both encryption and decryption latency, making it ideal for resource-constrained applications. Analyzing the distribution of neighboring pixels across all directions in both original and encrypted videos reveals that there is no correlation between the original and encrypted frames. Adjacent pixels must be highly decorrelated in all directions in order to fight against statistical attacks. For video encryption, our approach produces lower correlation coefficients. Consequently, our system remains resistant to statistical attacks. For improved security, the histogram of an encrypted video frame must have a uniform distribution. We verify that the histogram of encrypted frames is uniformly distributed. So, our suggested strategy can protect against a variety of statistical attacks. We compute the Number of Bit Change Rate (NBCR) between video frames in order to quantify plaintext sensitivity. NBCR is close to 50% when the sequences are statistically independent. In our scheme, we get an NBCR of around 50% by changing a single bit in the plaintext. Two encrypted reference frames of the Coastguard video are shown in Fig.6.11, which vary in the plaintext by only one bit. The two encrypted frames are very different and have uniform histograms despite this little alteration. Additionally, we show the differences between these two encrypted frames and discover that the difference result provides a uniform distribution. Because of the bidirectional substitution-based avalanche effect, our technique is very resistant to differential attacks. As shown in Fig.6.12, even a little alteration in the secret key inhibits proper decryption, resulting in a recovered frame that is completely different from the original.

6.5 Disscussion

In this work, we proposed a unified video compression and encryption scheme for resource-constrained WVSNs using adaptive parallel compressive sensing. We suggested the LLC map, which is used in APCS and encryption. The LLC map has higher LE, SE, and unpredictability. In our scheme, frames within each shot of the video are marked as reference and non-reference frames, and then sampled at different sampling rates. An LLC map is used to initialize measurement matrices, which are subsequently modified using QR decomposition via Givens rotations to

provide better recovery quality. We encrypt the compressed sample using a swap-based permutation, followed by a novel bidirectional substitution mechanism where operations are selected from chaotic sequences. In the future, we aim to develop a combined compression and encryption framework that can be expanded by integrating lightweight deep learning models capable of adapting to dynamic sensor settings, reducing computational costs, and improving security [155].

6.6 Appendix

The complete derivation of the Jacobian $J(x)$ of the proposed LLC chaotic map is provided here for clarity.

$$x_{i+1} = f(x_i) = \frac{1000\,|\log(ax(1-x))|}{|\log(x^3 - bx)|} \tag{6.20}$$

Let

$$u(x) = 1000\,|\log(ax(1-x))|\,, \qquad v(x) = \left|\log\left(x^3 - bx\right)\right| \tag{6.21}$$

so that

$$f(x) = \frac{u(x)}{v(x)}. \tag{6.22}$$

Applying the quotient rule gives

$$f'(x) = \frac{u'(x)v(x) - u(x)v'(x)}{v(x)^2}. \tag{6.23}$$

Using the derivative of the absolute value function,

$$\frac{d}{dx}|g(x)| = \operatorname{sgn}(g(x))\, g'(x), \tag{6.24}$$

the derivative of $u(x)$ becomes

$$u'(x) = 1000\ \operatorname{sgn}(\log(ax(1-x)))\,\frac{1-2x}{x(1-x)}. \tag{6.25}$$

Similarly, the derivative of $v(x)$ is

$$v'(x) = \operatorname{sgn}\big(\log\big(x^3 - bx\big)\big)\,\frac{3x^2 - b}{x^3 - bx}. \tag{6.26}$$

Substituting (6.25) and (6.26) into (6.23), we obtain

$$\begin{aligned} f'(x) = \frac{1000}{\left|\log(x^3 - bx)\right|^2}\Bigg[&\left|\log\big(x^3 - bx\big)\right| \operatorname{sgn}(\log(ax(1-x)))\,\frac{1-2x}{x(1-x)} \\ &- \left|\log(ax(1-x))\right| \operatorname{sgn}\big(\log\big(x^3 - bx\big)\big)\,\frac{3x^2 - b}{x^3 - bx}\Bigg]. \end{aligned} \tag{6.27}$$

Evaluating at the fixed point x^*, the Jacobian is

$$J(x^*) = f'(x^*). \tag{6.28}$$

Therefore,

$$\begin{aligned} J(x^*) = \frac{1000}{\left|\log((x^*)^3 - bx^*)\right|^2}\Bigg[&\left|\log\big((x^*)^3 - bx^*\big)\right| \operatorname{sgn}(\log(ax^*(1-x^*)))\,\frac{1-2x^*}{x^*(1-x^*)} \\ &- \left|\log(ax^*(1-x^*))\right| \operatorname{sgn}\big(\log\big((x^*)^3 - bx^*\big)\big)\,\frac{3(x^*)^2 - b}{(x^*)^3 - bx^*}\Bigg]. \end{aligned} \tag{6.29}$$

Chapter 7

Conclusion

This thesis investigates video compression and encryption techniques using chaotic maps. The described methodologies have been evaluated through qualitative and quantitative analysis, illustrating their effectiveness. In this chapter, we conclude the dissertation by summarizing the key contributions and limitations of the work, followed by a discussion of potential directions for future research.

7.1 Concluding Remarks

In this thesis, we have proposed efficient video processing schemes for compression and security through a progressive sequence of contributions spanning uncompressed video encryption, deep compressive sensing-based video compression, joint video compression and encryption, and an APCS-based unified framework for shot-level secure video processing in resource-constrained environments. The overall outcome of the thesis shows that video security and video compression should not be treated as completely independent problems, particularly for applications that demand low computational complexity, high recovery quality, and robustness against attacks. Instead, the work establishes that better performance can be achieved through coordinated advances in chaotic-map design, temporal redundancy exploitation, deep reconstruction, adaptive sampling, efficient measurement-matrix design, and integrated compression–encryption strategies.

The first important finding of the thesis is that plaintext-dependent chaotic sequence generation together with carefully designed permutation–diffusion mechanisms can significantly improve the security of uncompressed video encryption. In the frame-level scheme, a new chaotic sequence is generated for each frame and coupled with secure sort-index-based permutation and block-level diffusion. Ex-

perimental analysis confirmed strong resistance against brute-force, key-sensitivity, differential, known-plaintext, and chosen-plaintext attacks, thereby establishing the effectiveness of the proposed design for secure uncompressed video encryption.

The second important finding is that exploiting temporal redundancy at the shot level can substantially improve encryption efficiency without degrading security. By processing video in a divide-and-conquer manner and considering thresholded difference-frame information, the shot-level scheme reduces the amount of data to be encrypted while preserving strong protection. The proposed LT-ICM chaotic map further improves unpredictability and security strength. The experimental results demonstrated improvements in encryption time, entropy, correlation coefficient, PSNR, and SSIM, thereby confirming that temporal redundancy can be effectively used to design faster and more efficient video encryption schemes.

The third major finding is that deep compressive sensing can provide improved reconstructed video quality at low sampling ratios compared to conventional compressive sensing approaches. In the proposed GOP-level deep CS framework, high-quality recovery is achieved by combining initial recovery with deep recovery based on multilevel feature compensation from keyframes and neighbouring non-keyframes. The use of an SSIM-based loss function further enhances recovered visual quality. Comparisons with state-of-the-art methods showed superior performance in terms of PSNR and SSIM, demonstrating the importance of jointly exploiting spatial and temporal information in deep video reconstruction.

Subsequently, we proposed a novel joint video compression and encryption framework in which an innovative permutation operation is performed prior to parallel compressive sensing using an improved chaotic map. The proposed permutation enhances the reconstruction quality of the recovered video, while the new substitution mechanism and plaintext-dependent chaotic sequence-based shuffling significantly improve the security of the video frames. Comparative and experimental evaluations with existing methods confirm the effectiveness of the proposed framework.

In the subsequent work, we developed a robust and secure framework for joint video compression and encryption with high-quality recovery. The proposed scheme integrates deep compressive sensing for video compression with DSDX-based encryption for secure protection of the compressed data. Through multilevel feature compensation involving both keyframes and adjacent non-keyframes, the method enhances the average quality of the recovered video frames, particularly at low sampling ratios. Experimental comparisons with state-of-the-art deep learning-based image and video compressive sensing methods showed that the proposed approach

achieves superior reconstruction quality in terms of PSNR and SSIM. Moreover, the proposed encryption model demonstrated improved performance over existing image and video encryption schemes in terms of mean correlation coefficient, NPCR, UACI, and entropy.

The final and most practically significant finding of this thesis is that adaptive parallel compressive sensing combined with improved chaotic-map-based encryption provides an effective framework for secure video processing in resource-constrained environments. In the proposed shot-level APCS scheme, different sampling rates are assigned according to motion characteristics, and LLC-map-based measurement matrices are refined through QR decomposition using Givens rotations to improve recovery quality. The same chaotic sequence is also used to control permutation and substitution operations in encryption. The results demonstrated superior performance in terms of PSNR, SSIM, decompression time, entropy, NPCR, key space, and correlation-based security measures. This establishes the suitability of the proposed framework for real-time and low-resource applications such as Wireless Visual Sensor Networks (WVSNs).

7.2 Future Directions

Although the proposed schemes have shown efficiency in both compression and security, several promising research directions still remain. With the rise of neural and learned video compression, techniques like ROI-based Gaussian splatting [156] and end-to-end learned codecs [157] can greatly enhance real-time performance and adapt to hardware limitations. On the security front, implicit neural representation–based encryption such as NeR-VCP [158] opens new opportunities for combining compression with strong cryptographic protection. Additionally, adaptive compression parameter optimization for video analytics [159] offers a way to maintain recognition accuracy while reducing data size. Exploring these paths could lead to sustainable, low-latency, and intelligent video processing systems that are ideal for next-generation applications such as real-time surveillance, cloud-edge computing, and immersive media.

In the context of compressive sensing (CS), several emerging directions can further enhance video compression and security. Real-time mobile implementations such as MobileSCI demonstrate that efficient architectures can achieve high-quality video reconstruction at approximately 35 FPS on commercial devices, making CS highly practical for on-device video processing [160]. Adaptive frameworks like block-

based sampling-rate control adjust measurement rates dynamically based on scene changes, reducing redundancy and improving bandwidth efficiency while maintaining reconstruction quality [161]. From a security perspective, discretization-enhanced CS approaches such as SECVID show strong potential in defending against adversarial video attacks by dispersing malicious perturbations into the sparse domain, thereby improving classification robustness across standard benchmarks [162]. In addition, extensive analyses highlight the importance of refining sampling methodologies, improving measurement encoding, and improving reconstruction techniques. Together, these advancements offer a blueprint for the development of future CS-based codecs [163]. Collectively, these advancements imply that the incorporation of adaptive, secure, and hardware-conscious CS frameworks is crucial for the development of sustainable, real-time, and robust video processing systems, applicable in fields such as wireless sensor networks and cloud-edge computing.

Bibliography

[1] C. Xiao, L. Wang, M. Zhu, and W. Wang, "A resource-efficient multimedia encryption scheme for embedded video sensing system based on unmanned aircraft," *Journal of Network and Computer Applications*, vol. 59, pp. 117–125, 2016.

[2] J. Karmakar, A. Pathak, D. Nandi, and M. K. Mandal, "Sparse representation based compressive video encryption using hyper-chaos and dna coding," *Digit. Signal Process.*, vol. 117, p. 103143, 2021.

[3] A. Pande, P. Mohapatra, and J. Zambreno, "Securing multimedia content using joint compression and encryption," *IEEE MultiMedia*, vol. 20, no. 4, pp. 50–61, 2012.

[4] J. Sethi, J. Bhaumik, and A. S. Chowdhury, "Joint video compression and encryption using parallel compressive sensing and improved chaotic maps," *Digit. Signal Process.*, vol. 130, p. 103746, 2022.

[5] A. Uhl and A. Pommer, *Image and video encryption: from digital rights management to secured personal communication*. Springer Science & Business Media, 2004, vol. 15.

[6] S. Lian, *Multimedia content encryption: techniques and applications*. Auerbach Publications, 2008.

[7] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital image and video," in *Multimedia Encryption and Authentication Techniques and Applications*. Auerbach Publications, 2006, pp. 129–163.

[8] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International journal of bifurcation and chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.

[9] J. Sethi, J. Bhaumik, and A. S. Chowdhury, "Fast and secure video encryption using divide-and-conquer and logistic tent infinite collapse chaotic map," in *Proc. Int. Conf. Comput. Vis. Image Process. (CVIP)*. Springer, 2022, pp. 151–163.

[10] H. Xu, X. Tong, Z. Wang, M. Zhang, Y. Liu, and J. Ma, "Robust video encryption for h. 264 compressed bitstream based on cross-coupled chaotic cipher," *Multimedia Systems*, vol. 26, no. 4, pp. 363–381, 2020.

[11] Z. Hua, F. Jin, B. Xu, and H. Huang, "2d logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[12] Z. Hua and Y. Zhou, "Image encryption using 2d logistic-adjusted-sine map," *Information Sciences*, vol. 339, pp. 237–253, 2016.

[13] H. Elkamchouchi, W. M. Salama, and Y. Abouelseoud, "New video encryption schemes based on chaotic maps," *IET Image Process.*, vol. 14, no. 2, pp. 397–406, 2019.

[14] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Information Sciences*, vol. 480, pp. 403–419, 2019.

[15] H. Gan, S. Xiao, Y. Zhao, and X. Xue, "Construction of efficient and structural chaotic sensing matrix for compressive sensing," *Signal Processing: Image Communication*, vol. 68, pp. 129–137, 2018.

[16] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[17] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[18] Z. Ke, W. Huang, Z.-X. Cui, J. Cheng, S. Jia, H. Wang, X. Liu, H. Zheng, L. Ying, Y. Zhu *et al.*, "Learned low-rank priors in dynamic mr imaging," *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3698–3710, 2021.

[19] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor admm-net for snapshot compressive imaging," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 223–10 232.

[20] W. Shi, S. Liu, F. Jiang, and D. Zhao, "Video compressed sensing using a convolutional neural network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, pp. 425–438, 2 2021.

[21] J. Bigot, C. Boyer, and P. Weiss, "An analysis of block sampling strategies in compressed sensing," *IEEE transactions on information theory*, vol. 62, no. 4, pp. 2125–2139, 2016.

[22] Y. Wu, M. Rosca, and T. Lillicrap, "Deep compressed sensing," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6850–6860.

[23] H. Fang, S. A. Vorobyov, H. Jiang, and O. Taheri, "Permutation meets parallel compressed sensing: How to relax restricted isometry property for 2D sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 196–210, 2013.

[24] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Proc. Data Compress. Conf.*, Mar. 2011, pp. 183–192.

[25] M. Azghani, M. Karimi, and F. Marvasti, "Multihypothesis compressed video sensing technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 627–635, 2016.

[26] C. Chen, C. Zhou, P. Liu, and D. Zhang, "Iterative reweighted tikhonov-regularized multihypothesis prediction scheme for distributed compressive video sensing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 1–10, 2020.

[27] J. Wang and B. Shim, "On the recovery limit of sparse signals using orthogonal matching pursuit," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4973–4976, 2012.

[28] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[29] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. CVPR*, 2016, pp. 449–458.

[30] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME))*, 2017, pp. 877–882.

[31] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 375–388, 2020.

[32] F. Liu and H. Koenig, "A survey of video encryption algorithms," *Computers & Security*, vol. 29, no. 1, pp. 3–15, 2010.

[33] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognit.*, vol. 37, no. 4, pp. 725–737, 2004.

[34] A. S. Unde and P. Deepthi, "Design and analysis of compressive sensing-based lightweight encryption scheme for multimedia iot," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 1, pp. 167–171, 2019.

[35] V. Pudi, A. Chattopadhyay, and K.-Y. Lam, "Secure and lightweight compressive sensing using stream cipher," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 65, no. 3, pp. 371–375, 2017.

[36] J. Mou, L. Tan, Y. Cao, N. Zhou, and Y. Zhang, "Multi-face image compression encryption scheme combining extraction with stp-cs for face database," *IEEE Internet of Things Journal*, 2025.

[37] S. Lian, J. Sun, Z. Wang, and Y. Dai, "A fast video encryption scheme based-on chaos," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, vol. 1. IEEE, 2004, pp. 126–131.

[38] D. Socek, S. Magliveras, D. Ćulibrk, O. Marques, H. Kalva, and B. Furht, "Digital video encryption algorithms based on correlation-preserving permutations," *EURASIP journal on Information Security*, vol. 2007, pp. 1–15, 2007.

[39] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing: Image Communication*, vol. 23, no. 3, pp. 212–223, 2008.

[40] D. Valli and K. Ganesan, "Chaos based video encryption using maps and ikeda time delay system," *The European Physical Journal Plus*, vol. 132, pp. 1–18, 2017.

[41] F. Liu and H. Koenig, "Puzzle–a novel video encryption algorithm," in *IFIP International Conference on Communications and Multimedia Security*. Springer, 2005, pp. 88–97.

[42] H. Kezia and G. F. Sudha, "Encryption of digital video based on lorenz chaotic system," in *2008 16th International Conference on Advanced Computing and Communications*. IEEE, 2008, pp. 40–45.

[43] R. N. Hole and M. Kolhekar, "Robust encryption of uncompressed videos with a selective frame scheme," in *2018 3rd International Conference for Convergence in Technology (I2CT)*. IEEE, 2018, pp. 1–7.

[44] X. Zhang, S.-H. Seo, and C. Wang, "A lightweight encryption method for privacy protection in surveillance videos," *IEEE Access*, vol. 6, pp. 18 074–18 087, 2018.

[45] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on signal processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[46] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

[47] C. N. Raju, G. Umadevi, K. Srinathan, and C. Jawahar, "Fast and secure real-time video encryption," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 257–264.

[48] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using scan patterns," *Pattern recognition*, vol. 37, no. 4, pp. 725–737, 2004.

[49] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for IoT systems using probabilistic image encryption," *IEEE Trans Ind Inform*, vol. 14, no. 8, pp. 3679–3689, 2018.

[50] S. Li, X. Zheng, X. Mou, and Y. Cai, "Chaotic encryption scheme for real-time digital video," in *Real-Time Imaging VI*, vol. 4666. SPIE, 2002, pp. 149–160.

[51] K. Ganesan, I. Singh, and M. Narain, "Public key encryption of images and videos in real time using chebyshev maps," in *2008 Fifth International Conference on Computer Graphics, Imaging and Visualisation*. IEEE, 2008, pp. 211–216.

[52] D. Ganeshkumar, A. Suresh, K. Manigandan *et al.*, "A new one round video encryption scheme based on 1d chaotic maps," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019, pp. 439–444.

[53] H. Elkamchouchi, W. M. Salama, and Y. Abouelseoud, "New video encryption schemes based on chaotic maps," *IET Image Processing*, vol. 14, no. 2, pp. 397–406, 2020.

[54] G. Marsaglia, "Diehard statistical tests," Online. [Online]. Available: http://stat.fsu.edu/pub/diehard/

[55] R. Ranjithkumar, D. Ganeshkumar, A. Suresh, and K. Manigandan, "A new one round video encryption scheme based on 1D chaotic maps," in *5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019, pp. 439–444.

[56] Y. Wu, J. P. Noonan, S. Agaian *et al.*, "Npcr and uaci randomness tests for image encryption," *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[57] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3584–3592.

[58] J. Sethi, J. Bhaumik, and A. S. Chowdhury, "Chaos-based uncompressed frame level video encryption," in *Proceedings of the Seventh International Conference on Mathematics and Computing: ICMC 2021*. Springer, 2022, pp. 201–217.

[59] A. A. Abd El-Latif, B. Abd-El-Atty, W. Mazurczyk, C. Fung, and S. E. Venegas-Andraca, "Secure data encryption based on quantum walks for 5g internet of things scenario," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 118–131, 2020.

[60] X.-H. Song, H.-Q. Wang, S. E. Venegas-Andraca, and A. A. Abd El-Latif, "Quantum video encryption based on qubit-planes controlled-xor operations and improved logistic map," *Physica A: Statistical Mechanics and its Applications*, vol. 537, p. 122660, 2020.

[61] D. He, C. He, L.-G. Jiang, H.-w. Zhu, and G.-r. Hu, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 7, pp. 900–906, 2001.

[62] Q. Huynh-Thu and M. Ghanbari, "The accuracy of psnr in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, pp. 35–48, 2012.

[63] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[64] R. Monika and S. Dhanalakshmi, "An efficient medical image compression technique for telemedicine systems," *Biomedical Signal Processing and Control*, vol. 80, p. 104404, 2023.

[65] S. Zhang, Y. Lin, and Q. Liu, "Secure and efficient video surveillance in cloud computing," in *IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, 2014, pp. 222–226.

[66] Y. Piao, K. Choi, K. P. Choi, M. Park, and M. W. Park, "Depth-wise split unit coding order for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6375–6384, 2022.

[67] M. A. Yılmaz and A. M. Tekalp, "End-to-end rate-distortion optimized learned hierarchical bi-directional video compression," *IEEE Trans. Image Process.*, vol. 31, pp. 974–983, 2022.

[68] S. Pudlewski, A. Prasanna, and T. Melodia, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mob. Comput.*, vol. 11, no. 6, pp. 1060–1072, 2012.

[69] N. Vaswani, "LS-CS-Residual (LS-CS): Compressive sensing on least squares residual," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4108–4120, 2010.

[70] C. LI, W. YIN, and Y. ZHANG, "Tval3: Tv minimization by augmented lagrangian and alternating direction algorithms. [online]. available: http://www.caam.rice.edu/ optimization/l1/tval3/." 2009.

[71] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, pp. 3336–3351, 2014.

[72] M. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. Picture Coding Symp.*, vol. 1, no. 13, 2006.

[73] S. G. Lingala, Y. Hu, E. Dibella, and M. Jacob, "Accelerated dynamic mri exploiting sparsity and low-rank structure: K-t slr," *IEEE Trans. Med. Imag.,*, vol. 30, pp. 1042–1054, 5 2011.

[74] C. Zhao, S. Ma, J. Zhang, R. Xiong, and W. Gao, "Video compressive sensing reconstruction via reweighted residual sparsity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, pp. 1182–1195, 6 2017.

[75] E. W. Tramel and J. E. Fowler, "Video compressed sensing with multihypothesis," in *Proc. Data Compress. Conf.*, 2011, pp. 193–202.

[76] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.

[77] R. Nan, G. Sun, B. Zheng, and L. Wang, "Adaptive deep learning network for image reconstruction of compressed sensing," *Signal, Image and Video Processing*, vol. 18, no. 2, pp. 1463–1475, 2024.

[78] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans Comput. Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[79] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*. Association for Computing Machinery, Inc, 10 2015, pp. 689–692.

[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 12 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[81] K. Q. Dinh, T. N. Canh, and B. Jeon, "Compressive sensing of video with weighted sensing and measurement allocation," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2065–2069.

[82] F. Liu and H. Koenig, "A survey of video encryption algorithms," *Comput. & Secur.*, vol. 29, no. 1, pp. 3–15, 2010.

[83] A. S. Unde and P. Deepthi, "Design and analysis of compressive sensing-based lightweight encryption scheme for multimedia IoT," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 67, no. 1, pp. 167–171, 2019.

[84] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "A performance comparison of measurement matrices in compressive sensing," *Int. J. of Commun. Syst.*, vol. 31, no. 10, p. e3576, 2018.

[85] X.-H. Song, H.-Q. Wang, S. E. Venegas-Andraca, and A. A. Abd El-Latif, "Quantum video encryption based on qubit-planes controlled-XOR operations and improved logistic map," *Phys. A: Stat. Mech. Appl.*, vol. 537, p. 122660, 2020.

[86] X. Zhang, S.-H. Seo, and C. Wang, "A lightweight encryption method for privacy protection in surveillance videos," *IEEE Access*, vol. 6, pp. 18 074–18 087, 2018.

[87] A. A. Abd El-Latif, B. Abd-El-Atty, W. Mazurczyk, C. Fung, and S. E. Venegas-Andraca, "Secure data encryption based on quantum walks for 5G internet of things scenario," *IEEE Trans. on Netw. Serv. Manage.*, vol. 17, no. 1, pp. 118–131, 2020.

[88] D. He, C. He, L.-G. Jiang, H.-w. Zhu, and G.-r. Hu, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, vol. 48, no. 7, pp. 900–906, 2001.

[89] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for IoT systems using probabilistic image encryption," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3679–3689, 2018.

[90] N. Khlif, A. Masmoudi, F. Kammoun, and N. Masmoudi, "Secure chaotic dual encryption scheme for H.264/AVC video conferencing protection," *IET Image Process.*, vol. 12, no. 1, pp. 42–52, 2018.

[91] J. Karmakar, A. Pathak, D. Nandi, and M. K. Mandal, "Sparse representation based compressive video encryption using hyper-chaos and dna coding," *Digit. Signal Process.*, vol. 117, p. 103143, 2021.

[92] C. Chen, C. Zhou, P. Liu, and D. Zhang, "Iterative reweighted tikhonov-regularized multihypothesis prediction scheme for distributed compressive video sensing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 1–10, 2018.

[93] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends in Signal Processing*, vol. 4, pp. 297–416, 2012.

[94] J. Chen, Y. Chen, D. Qin, and Y. Kuo, "An elastic net-based hybrid hypothesis method for compressed video sensing," *Multimed. Tools. Appl.*, vol. 74, no. 6, pp. 2085–2108, 2015.

[95] M. Azghani, M. Karimi, and F. Marvasti, "Multihypothesis compressed video sensing technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 627–635, 2015.

[96] J. Karmakar, D. Nandi, and M. K. Mandal, "A novel hyper-chaotic image encryption with sparse-representation based compression," *Multimed. Tools. Appl.*, vol. 79, no. 37, pp. 28 277–28 300, 2020.

[97] P. Ping, X. Yang, X. Zhang, Y. Mao, and H. Khalid, "Generating visually secure encrypted images by partial block pairing-substitution and semi-tensor product compressed sensing," *Digit. Signal Process.*, vol. 120, p. 103263, 2022.

[98] L. Yu, J. P. Barbot, G. Zheng, and H. Sun, "Compressive sensing with chaotic sequence," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 731–734, 2010.

[99] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[100] E. Candes and J. Romberg, "l1-magic: Recovery of sparse signals via convex programming," *URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf*, vol. 4, no. 14, p. 16, 2005.

[101] Y. Liu, T. Lin, J. Wang, and H. Yuan, "Bit image encryption algorithm based on hyper chaos and dna sequence," *J. Comput.*, vol. 29, no. 3, pp. 43–55, 2018.

[102] V. Lappalainen, A. Hallapuro, and T. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 717–725, 2003.

[103] H. Kwok and W. K. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos Solitons Fractals*, vol. 32, no. 4, pp. 1518–1529, 2007.

[104] L. Yang, Z. Han, Z. Huang, and J. Ma, "A remotely keyed file encryption scheme under mobile cloud computing," *J. Netw. Comput. Appl.*, vol. 106, pp. 90–99, 2018.

[105] R. Li, H. Liu, R. Xue, and Y. Li, "Compressive-sensing-based video codec by autoregressive prediction and adaptive residual recovery," *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 8, p. 562840, 2015.

[106] J. Sethi, M. Mondal, J. Bhaumik, and A. S. Chowdhury, "Deep compressive sensing for high-quality video recovery," in *Proc. Int. Conf. Computer Vision and Image Process.(CVIP), 2024*. Springer, (2024), pp. 119–134.

[107] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 01 2002.

[108] L. Chen, H. Yin, L. Yuan, J. T. Machado, R. Wu, and Z. Alam, "Double color image encryption based on fractional order discrete improved Henon map and Rubik's cube transform," *Signal Process. Image Commun.*, vol. 97, p. 116363, 2021.

[109] K. Shahna and A. Mohamed, "Novel hyper chaotic color image encryption based on pixel and bit level scrambling with diffusion," *Signal Process. Image Commun.*, vol. 99, p. 116495, 2021.

[110] K. Meng and Y. Wo, "An image compression and encryption scheme for similarity retrieval," *Signal Process. Image Commun.*, vol. 119, p. 117044, 2023.

[111] H. Lee, K. Lee, and Y. Shin, "Implementation and performance analysis of aes-128 cbc algorithm in wsns," in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 1. IEEE, 2010, pp. 243–248.

[112] J. Wu, X. Liao, and B. Yang, "Image encryption using 2d hénon-sine map and dna approach," *Signal Process.*, vol. 153, pp. 11–23, 2018.

[113] Q. Lai, H. Hua, X.-W. Zhao, U. Erkan, and A. Toktas, "Image encryption using fission diffusion process and a new hyperchaotic map," *Chaos, Solitons & Fractals*, vol. 175, p. 114022, 2023.

[114] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Phys. Rev. Lett.*, vol. 88, no. 17, p. 174102, 2002.

[115] S. He, K. Sun, and H. Wang, "Complexity analysis and dsp implementation of the fractional-order lorenz hyperchaotic system," *Entropy*, vol. 17, no. 12, pp. 8299–8311, 2015.

[116] M. Sharma, R. K. Ranjan, and V. Bharti, "A pseudo-random bit generator based on chaotic maps enhanced with a bit-xor operation," *J. Inform. Secur. Appl.*, vol. 69, p. 103299, 2022.

[117] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digital Signal Processing*, vol. 72, pp. 9–18, 2018.

[118] H. Kwok and W. K. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp. 1518–1529, 2007.

[119] G. Hu, D. Xiao, Y. Wang, and T. Xiang, "An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 116–127, 2017.

[120] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2d compressive sensing," *Optics & Laser Technology*, vol. 82, pp. 121–133, 2016.

[121] W. Hao, T. Zhang, X. Chen, and X. Zhou, "A hybrid neqr image encryption cryptosystem using two-dimensional quantum walks and quantum coding," *Signal Processing*, vol. 205, p. 108890, 2023.

[122] C. Qiu and X. Hu, "Adacs: Adaptive compressive sensing with restricted isometry property-based error-clamping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46(7), p. 4702–4719, 2024.

[123] Z. Wang and F. Wang, "Wireless visual sensor networks: Applications, challenges, and recent advances," in *2019 SoutheastCon*, 2019, pp. 1–8.

[124] J. Zhang, P.-W. Tsai, X. Xue, X. Ye, and S. Zhang, "A comprehensive data gathering network architecture in large-scale visual sensor networks," *Plos one*, vol. 15, no. 1, p. e0226649, 2020.

[125] T. Pal and S. DasBit, "A low-overhead secure image compression over wireless multimedia sensor network," in *Proc. IEEE Int. Symp. Smart Electron. Syst*, 2019, pp. 1–6.

[126] G. Sudha and C. Tharini, "Analysis of wavelets on discrete wavelet transform for image compression and transmission in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun., Comput. Internet Things*, 2022, pp. 1–5.

[127] C. Han, S. Zhang, B. Zhang, J. Zhou, and L. Sun, "A distributed image compression scheme for energy harvesting wireless multimedia sensor networks," *Sensors*, vol. 20, no. 3, p. 667, 2020.

[128] G. L. Priya and D. Ghosh, "An effectual video compression scheme for wvsns based on block compressive sensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 1542–1552, 2024.

[129] N. Zidani, N. Kouadria, N. Doghmane, and S. Harize, "Low complexity pruned dct approximation for image compression in wireless multimedia sensor networks," in *Proc. IEEE 5th Int. Conf. Front. Signal Process.*, 2019, pp. 26–30.

[130] Y. Tian and R. Liao, "Multinode collaborative image compression algorithm for wireless multimedia sensor networks based on lbt," *IEEE Sensors J.*, vol. 20, no. 20, pp. 12 065–12 073, 2020.

[131] J. Uthayakumar, M. Elhoseny, and K. Shankar, "Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in wsn," *IEEE Trans. on Reliab.*, vol. 69, no. 4, pp. 1398–1423, 2020.

[132] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in wireless sensor networks," *Future Generation Computer Systems*, vol. 64, pp. 151–162, 2016.

[133] X. Yuan and R. Haimi-Cohen, "Image compression based on compressive sensing: End-to-end comparison with jpeg," *IEEE Trans. on Multimedia*, vol. 22, no. 11, pp. 2889–2904, 2020.

[134] J. Zhang, Q. Xiang, Y. Yin, C. Chen, and X. Luo, "Adaptive compressed sensing for wireless image sensor networks," *Multimed. Tools Appl.*, vol. 76, no. 3, pp. 4227–4242, 2017.

[135] M. Ebrahim, S. H. Adil, T. Gul, and K. Raza, "Comparative analysis: Conventional video codecs v/s compressive sensing video codecs," in *Proc. IEEE 3rd Int. Conf. Emerg. Trends Eng., Scie. Technol.* IEEE, 2018, pp. 1–6.

[136] S. Mun and J. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. 16th IEEE Int. Conf. Image Process.(ICIP)*, 2009, pp. 3021–3024.

[137] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *2011 Data Compression Conference*, 2011, pp. 183–192.

[138] R. Wu, S. Gao, X. Wang, S. Liu, Q. Li, U. Erkan, and X. Tang, "Aea-ncs: An audio encryption algorithm based on a nested chaotic system," *Chaos, Solitons & Fract.*, vol. 165, p. 112770, 2022.

[139] L. Teng, P. Cao, and Y. Liu, "Multi-image encryption algorithm based on novel spatiotemporal chaotic system and dynamical chaotic trajectories," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 2, pp. 1562–1575, 2025.

[140] M. Gong, X. Chai, Y. Lu, and Y. Zhang, "Exploiting four-dimensional chaotic systems with dissipation and optimized logical operations for secure image compression and encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7628–7642, 2024.

[141] J. Li and X. Hou, "The design of an adaptive enhanced amp-based image block compressed sensing and its application to image encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 11, pp. 11 128–11 141, 2024.

[142] Q. Lai and G. Hu, "A nonuniform pixel split encryption scheme integrated with compressive sensing and its application in iomt," *IEEE Trans. on Ind. Informat.*, vol. 20, no. 9, pp. 11 262–11 272, 2024.

[143] Z. Hua, K. Zhang, Y. Li, and Y. Zhou, "Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing," *Signal Process.*, vol. 183, p. 107998, 2021.

[144] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotification system for improving chaos complexity," *IEEE Transactions on signal processing*, vol. 68, pp. 1937–1949, 2020.

[145] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: nonlinear phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[146] Z. Hua and Y. Zhou, "Exponential chaotic model for generating robust chaos," *IEEE Trans. Syst., Man, Cybern. Syst*, vol. 51, no. 6, pp. 3713–3724, 2019.

[147] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Am. J. Physiol. Heart Circ. Physiol.*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[148] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. Levenson, M. Vangel, N. A. Heckert, and D. L. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.

[149] D. Olteanu, N. Vortmeier, and D. Zivanović, "Givens rotations for QR decomposition, SVD and PCA over database joins," *The VLDB Journal*, vol. 33, no. 4, pp. 1013–1037, 2024.

[150] A. George and J. W. Liu, "Householder reflections versus givens rotations in sparse orthogonal decomposition," *Linear Algebra Appl.*, vol. 88, pp. 223–238, 1987.

[151] L. Teng, X. Wang, and Y. Xian, "Image encryption algorithm based on a 2d-clss hyperchaotic map using simultaneous permutation and diffusion," *Inf. Sci.*, vol. 605, pp. 71–85, 2022.

[152] Z. Gan, X. Chai, J. Zhang, Y. Zhang, and Y. Chen, "An effective image compression–encryption scheme based on compressive sensing (cs) and game of life (gol)," *Neural Comput. and Appl.*, vol. 32, no. 17, pp. 14 113–14 141, 2020.

[153] S. Zhang, X. Peng, X. Wang, C. Chen, and Z. Zeng, "A novel memristive multiscroll multistable neural network with application to secure medical image communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 2, pp. 1774–1786, 2025.

[154] D. Ding, D. Xie, H. Zhang, Z. Yang, and C. Liu, "Deepface-based chaotic image encryption using key optimization and semi-tensor product theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 7, pp. 6522–6534, 2025.

[155] N. Shylashree, S. Kumar, and H. Min, "Combined compression and encryption of linear wireless sensor network data using autoencoders," *Scientific Reports*, vol. 15, no. 1, p. 17771, 2025.

[156] L. Gupta and I. N. Junejo, "Neural video compression using 2d gaussian splatting," *arXiv preprint arXiv:2505.09324*, 2025.

[157] J. S. Gomes, M. Grellert, F. L. Ramos, and S. Bampi, "End-to-end neural video compression: A review," *IEEE Open Journal of Circuits and Systems*, 2025.

[158] Y. Lin, Y. Ke, K. Niu, J. Liu, and X. Yang, "Ner-vcp: A video content protection method based on implicit neural representation," *arXiv preprint arXiv:2408.15281*, 2024.

[159] K. Masykuroh, E. Mulyana, F. Krishna *et al.*, "A survey on video compression optimization techniques for accuracy enhancement in video analytics applications (vaps)," *IEEE Access*, 2025.

[160] M. Cao, L. Wang, H. Wang, G. Wang, and X. Yuan, "Towards real-time video compressive sensing on mobile devices," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 11 080–11 088.

[161] K. Iwama, R. Morita, and J. Zhou, "Block based adaptive compressive sensing with sampling rate control," in *Proceedings of the 5th ACM International Conference on Multimedia in Asia*, 2023, pp. 1–7.

[162] W. Song, C. Cong, H. Zhong, and J. Xue, "Correction-based defense against adversarial video attacks via {Discretization-Enhanced} video compressive sensing," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 3603–3620.

[163] J. Zhou and J. Yang, "Compressive sensing in image/video compression: Sampling, coding, reconstruction, and codec optimization," *Information*, vol. 15, no. 2, p. 75, 2024.

Jagannath Sethi