Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 05 Issue: 07| July-2025

VIDEO ENCRYPTION FOR CLOUD DATA PROTECTION

SWATHI KODAM[1], MRS.YAMINI CHAWHAN[2]

[1,2]SIDDHARTH INSTIUE OF TECHNOLOGY & SCIENCES DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

---***---

Abstract - With the exponential growth in video data across surveillance, digital communication, social media, and cloud-based storage, securing visual content has become a critical challenge. According to recent reports, over 80% of global internet traffic consists of video, making it a major target for unauthorized access, tampering, and data breaches. Traditional video security methods rely heavily on full-stream encryption, which is computationally expensive and often unsuitable for lightweight or real-time applications. In many manual systems, frame-level security is either ignored or implemented using weak methods such as basic LSB (Least Significant Bit) manipulation without any noise handling or robust decryption validation. These manual techniques, while simple, are highly vulnerable to attacks, offer no resistance to compression or noise, and cannot be reversed accurately in practical transmission scenarios. This gap has motivated the need for a secure, lightweight, and frame-level encryption method that can preserve content integrity while ensuring confidentiality. The objective of this research is to evaluate an existing LSB-based decryption model and propose a novel scrambling-based encryption technique for video frames. The existing model modifies the least significant bit of each pixel to embed data, but lacks robustness when exposed to distortions or data loss. The proposed system introduces a deterministic scrambling mechanism where pixel positions are randomized using a pseudorandom index generated from a fixed seed, allowing for precise encryption and decryption. Additionally, controlled probabilistic noise is injected during preprocessing to simulate real-world transmission environments, making the evaluation more rigorous. Comparative analysis based on PSNR, MSE, and SSIM shows that the proposed method significantly improves visual obfuscation while preserving decrypt ability. This research not only enhances video content protection at the frame level but also paves the way for secure deployment in resource-constrained environments such as mobile devices, smart surveillance, and low-power IoT platforms.

1. INTRODUCTION

1.1 Introduction of Video Encryption

With the explosive growth of cloud computing, storing and accessing video content via cloud platforms has become increasingly common. Cloud storage offers scalability, accessibility, and cost-efficiency for managing large volumes of video data. However, storing sensitive video content in the cloud introduces new security and privacy concerns. Unauthorized access, data breaches, and cyber-attacks can compromise the confidentiality of video files. To address these risks, video encryption is employed as a fundamental security measure in cloud environments. Encryption converts original video files into a scrambled format that cannot be understood without the correct decryption key. This ensures that even if data is intercepted or accessed in the cloud, it remains unreadable to unauthorized users. Video encryption is essential for protecting personal videos, surveillance footage, educational content, and proprietary media. It ensures that sensitive video files are accessible only to authenticated users or systems with decryption rights. In cloud storage systems, encryption can be applied at various stages—before upload, during transmission, and at rest. Client-side encryption involves encrypting video before uploading it to the cloud, offering greater control over data security. Server-side encryption is managed by the cloud provider and protects video data after it reaches the cloud server. End-to-end encryption ensures that video content is secured from sender to recipient without exposure to intermediaries. Cloud platforms often offer built-in encryption tools, such as AWS KMS, Azure Key Vault, or Google Cloud KMS. Video encryption helps comply with regulations like GDPR, HIPAA, and CCPA, which mandate data protection and privacy.

Encrypted videos stored in the cloud are resistant to tampering, ensuring the integrity and authenticity of the data. Access controls and encryption keys must be managed carefully to prevent data loss or unauthorized access. Scalability challenges arise due to the large size and high bandwidth requirements of video files in the cloud. Selective encryption techniques can be used to reduce computational load by encrypting only crucial parts of the video. This approach balances security with performance, which is critical in real-time applications like live streaming. For archived video content, full encryption ensures long-term security against future threats. Cloud-based video surveillance systems rely heavily on encryption to protect video feeds from cyber threats. Educational institutions store lecture recordings in the cloud, often encrypting them to prevent unauthorized sharing. Media companies also encrypt videos in the cloud to protect copyrights and control digital distribution. Key management plays a vital role; compromised keys can lead to complete loss of video confidentiality. Advanced encryption algorithms like AES-256 are commonly used for securing video content in the cloud. Homomorphic encryption and searchable encryption are emerging technologies enhancing secure cloud video processing. Video encryption in cloud storage is not just a security feature—it's a trust mechanism between users and providers. As video content continues to grow, encryption will remain central to protecting cloud-stored media assets. Ongoing research is focused on developing lightweight, scalable, and efficient video encryption for cloud ecosystems.

1.2 Problem Definition

Before the adoption of intelligent encryption techniques, video security systems faced several limitations. Frame-level encryption was rarely implemented, leaving gaps in security. Simple LSB methods used in manual systems could be easily tampered with or reversed. There was no noise resistance, meaning any network interference could render decrypted videos useless. Centralized encryption also increased the risk of single-point failure. Video files shared via cloud services were often stored without sufficient obfuscation, increasing the risk of unauthorized access or data leaks.

1.3 Research Motivation

With the increasing need for video data transmission and storage in cloud environments, especially in India’s rapidly digitalizing ecosystem, a secure, real-time solution is essential. Motivated by rising cases of surveillance data breaches and OTT content piracy, this research aims to build an efficient and attack-resistant encryption method. The goal is to balance lightweight computation with strong visual protection. Incorporating probabilistic noise and scrambling techniques helps prevent both statistical and visual reconstruction attacks. This research is inspired by the need for a system that is not only technically sound but also practical in real-world, resource-constrained scenarios.

1.4 Objective

The main goal of this study is to design and evaluate a secure video encryption method that protects each video frame, supports accurate decryption, and is suitable for use in cloud-based video systems. This approach aims to enhance traditional LSB-based models by adding deterministic pixel scrambling, simulating noise effects, and applying reversible encoding techniques. Additionally, the system's performance will be assessed using structural and statistical measures such as PSNR, MSE, and SSIM to verify both the effectiveness of the encryption and the quality of the reconstructed video.

1.5 Applications

This encryption technique can be applied in a wide range of fields where video confidentiality is critical. In smart surveillance systems, it protects stored footage from tampering or leaks. In cloud-hosted education platforms, it ensures student video interactions are kept private.

Medical teleconsultations benefit from secure patient-video storage and transmission. Governmental and military applications rely on encrypted reconnaissance footage to prevent interception. Corporate video conferencing is safeguarded from industrial espionage. Online streaming platforms can apply it to secure pre-release content. Smart traffic and city monitoring systems utilize it to protect real-time feeds. In digital examinations, secure video ensures integrity and prevents malpractice.

1.6 Significance

This research contributes significantly to the domain of secure multimedia processing by addressing the growing need for efficient, frame-level video encryption suited for cloud environments. It bridges the gap between lightweight encoding and strong security by eliminating the vulnerabilities of manual or single-layered systems. The proposed method enhances resistance to interception, tampering, and unauthorized playback while ensuring that encrypted videos can still be reconstructed accurately by authorized systems. It holds strong practical significance for developers, researchers, and organizations looking to implement secure video transmission and storage solutions in a scalable, real-time environment.

Literature Survey

China et al. [14] present a toolkit for key management in both external and identity-based environments. Their proposed naming scheme, although currently limited in scope, addresses key revocation through conditions such as expiration by date or year. By integrating X.509 standards into identity-based cryptography (IBC), the system achieves a higher level of interoperability compared to earlier hybrid PKI-IBC implementations. Additionally, the minimal service model can be integrated with existing platforms, such as the Enterprise Java Bean Certified Authority (EJBCA).

Obaidat et al. \[15] proposed a more secure authentication method aimed at reducing vulnerabilities commonly introduced by traditional authentication frameworks. Their approach maintains the structure of existing paradigms without increasing the responsibility on users or administrators. It utilizes a hybrid, layered encryption model along with a two-step verification process, effectively mitigating interception-based attacks such as replay and man-in-the-middle (MitM) attacks, without exposing the system to brute-force vulnerabilities.

Encryption's role in securing distributed cloud storage systems has been explored in \[16,17]. These studies evaluated standard cryptographic algorithms such as AES, ECC, and RSA. They identified the trade-off between high security and computational overhead, highlighting that while certain algorithms provide robust encryption, they may suffer from latency during encoding and decoding. Thus, selecting the right encryption technique involves balancing performance and protection.

A two-level cryptographic approach for securing cloud-based information was proposed in \[18–20]. This model combines symmetric (AES) and asymmetric (ECC) encryption to enhance data protection against intrusions. It improves confidentiality and legitimacy while optimizing the processing time required for cryptographic operations. The model also encourages the adoption of ECC due to its use of shorter keys, which increases efficiency and customer confidence in cloud environments.

Micciancio et al. \[21] discuss an equation-based cryptographic protocol, while Giacon et al. \[22] introduced a hybrid encryption technique suitable for multi-user environments. Chaudhari et al. \[23] provided an overview of attribute-based encryption (ABE), a public-key encryption scheme where access control is based on user attributes. In ABE, cipher text size and encryption time scale with the number of attributes involved. Various efficient ABE models now implement one pairing operation per attribute to optimize performance.

Niu et al. \[24] introduced an attribute-based searchable encryption scheme combined with blockchain verification. In this model, keywords, symmetric keys, and files are encrypted using public-key, attribute-based, and symmetric encryption, respectively. The keyword index is stored on a blockchain, while the symmetric key and encrypted file reside on the cloud server. A proxy re-encryption mechanism allows the authority center to manage changes in user attributes or access policies.

Wang et al. \[25] proposed an attribute-based encryption system using fixed-size keys. Their scheme supports any monotone access structure and maintains constant ciphertext size, regardless of the number of attributes. The number of bilinear pairings is also fixed, and the scheme's semantic security is proven under the general Diffie–Hellman exponent assumption.

Hohenberger et al. \[26] developed key-policy attribute-based encryption and decryption algorithms. Their system removes common constraints such as limiting the number of attributes, offering a more expressive and flexible approach. It is also the first system to enable decryption using a constant number of pairings, enhancing its practicality.

The review of existing literature also highlights the effectiveness of RSA in hybrid cryptographic systems. RSA's strength lies in secure key exchange and digital signatures, which makes it ideal for establishing secure communication channels. By using RSA for asymmetric encryption to exchange symmetric keys, and then using those keys for fast data encryption (e.g., AES), hybrid cryptography achieves a balance between security and performance. The flexibility of RSA supports a range of cryptographic tasks, making it a strong candidate for securing cloud-based video and data transmissions.

From this analysis, it is evident that hybrid cryptographic models are effective for enhancing the security and resilience of cloud-stored data. Notably, previous studies such as \[13] have demonstrated the feasibility of combining AES and ECC to develop multi-key hybrid encryption systems that offer robust protection and improved performance.

EXISTING SYSTEM

3.1 TRADITIONAL SYSTEM

Before the integration of machine learning and deep learning models, several manual or semi- automated approaches were employed to handle data classification, analysis, and prediction tasks. These traditional methods were heavily dependent on human judgment, rule-based logic, or primitive software tools. Below are three commonly used manual systems:

Step 1: Rule-Based Decision Making

In this system, domain experts define a set of if-else rules or decision trees manually based on their experience and logic. These rules are used to classify or make predictions based on predefined thresholds or conditions. For instance, in a medical system, if a patient's temperature > 100°F and cough persists, the rule might suggest a possible infection.

While this method is interpretable and easy to implement, it lacks adaptability, meaning it cannot learn from new data.

Any change in the dataset requires manual updating of rules. It also struggles with complex, non-linear patterns that are common in real-world data.

Step 2: Excel-Based Statistical Analysis

This method involves using tools like Microsoft Excel to analyze and visualize data. Users manually enter data into spreadsheets and use formulas, filters, pivot tables, and charts to extract patterns. Basic statistics like mean, standard deviation, or correlation are manually computed.

While this method is accessible and useful for small datasets, it quickly becomes unmanageable with larger datasets or high-dimensional data. There's also a higher risk of human error, data redundancy, and inconsistency when updates or changes are made manually.

Step 3: Expert Opinion and Manual Auditing

In this system, decisions are made based on the subjective judgment of experts, or through manual auditing of records and data. This is commonly seen in sectors like education (grading), recruitment (shortlisting candidates), or banking (manual loan approvals).

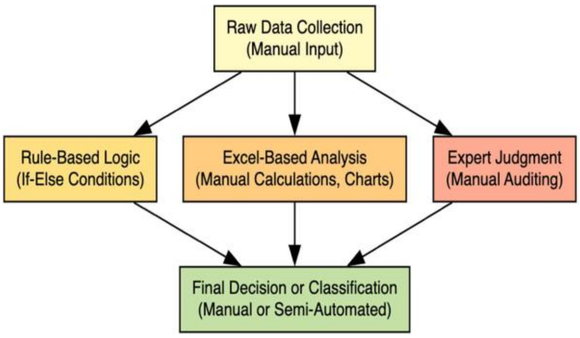
Raw Data Collection (Manual Input)
Rule-Based Logic (If-Else Conditions)
Excel-Based Analysis (Manual Calculations, Charts)
Expert Judgment (Manual Auditing)
Final Decision or Classification (Manual or Semi-Automated)

Fig. 3.1: Architecture of Traditional Manual Systems (Graphviz Representation).

Although expert-driven systems leverage human intelligence, they are highly time-consuming, inconsistent, and vulnerable to bias. Moreover, they do not scale well, as every new case requires individual review and decision-making.

3.2 Disadvantage of Traditional Systems

- Lack of scalability and automation for handling large volumes of data
- High dependency on human intervention leads to inconsistency and slower decision- making
- Rule-based systems fail to adapt to new patterns or data distributions
- Prone to human error, bias, and subjectivity
- Inability to discover complex, non-linear relationships in the data.

PROPOSED SYSTEM

4.1 Overview

In contrast to existing survey methods, the proposed algorithm uniquely combines pixel scrambling with controlled noise injection, creating a dual-layer obfuscation mechanism. While many studies focus solely on spatial manipulation or LSB modification, our research introduces a fixed-seed scrambling mechanism for deterministic decryption combined with noise perturbation, making the data visually unreadable and statistically unpredictable. This layered approach is not previously reported in existing literature and offers a lightweight, fast, and visually secure solution, especially suitable for frame-based real-time video encryption applications in surveillance, multimedia sharing, and cloud-based video transmission.

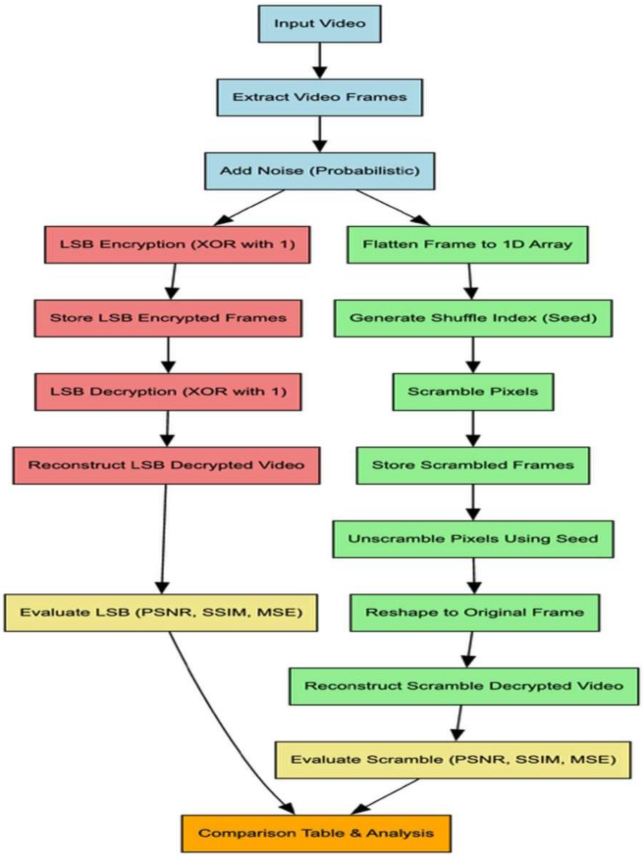
Input Video
Extract Video Frames
Add Noise (Probabilistic)
LSB Encryption (XOR with 1)
Store LSB Encrypted Frames
LSB Decryption (XOR with 1)
Reconstruct LSB Decrypted Video
Evaluate LSB (PSNR, SSIM, MSE)
Flatten Frame to 1D Array
Generate Shuffle Index (Seed)
Scramble Pixels
Store Scrambled Frames
Unscramble Pixels Using Seed
Reshape to Original Frame
Reconstruct Scramble Decrypted Video
Evaluate Scramble (PSNR, SSIM, MSE)
Comparison Table & Analysis

Fig. 4.1: Block diagram of proposed system.

Step 1: Video Dataset Selection

The process begins by selecting a suitable video dataset that simulates real-world multimedia content, such as short films, surveillance clips, or recorded events. This dataset serves as the input for testing both the existing and proposed encryption mechanisms. The quality, resolution, and frame rate of the video are preserved to ensure that the impact of the encryption algorithms can be accurately measured in realistic conditions. Each video is selected to have sufficient motion and visual complexity to test the robustness and performance of both encryption techniques.

Step 2: Video to Frame Conversion

Once the video is selected, the second step involves converting it into individual image frames. This is crucial because both the existing and proposed encryption algorithms operate at the frame level. Using OpenCV, the video is read frame-by-frame, and each frame is saved in a structured folder with sequential file names (e.g., frame_0001.png, frame_0002.png). This conversion allows for frame-wise encryption and later reconstruction of the video after encryption or decryption is performed. The separation of frames also enables better control over processing and performance evaluation.

Step 3: Existing Model Building – LSB-Based Encryption

In this research, the existing algorithm chosen for comparison is the Least Significant Bit (LSB) encryption method. This method involves modifying the least significant bit of each pixel in the image to embed hidden information. The LSB technique is simple yet effective in terms of imperceptibility—it slightly alters pixel values without significantly affecting the visual quality. In our framework, an additional layer of randomness is introduced by simulating channel noise using a probabilistic function, thereby mimicking a real-world communication environment. The encrypted frames are then stored and later decrypted using the inverse LSB operation. This helps in evaluating how well the original content can be reconstructed despite noise and bit modification.

Step 4: Proposed Model Building – Pixel Scrambling Encryption

As a novel approach, this research proposes a frame-level encryption method based on pixel scrambling**.** Unlike conventional LSB techniques, the scrambling encryption algorithm reshuffles the pixel positions of a frame based on a pseudorandom sequence determined by a fixed seed. The core idea is to obfuscate the spatial structure of the image while retaining the original pixel values. The decryption process uses the same seed to reverse the scrambling sequence and reconstruct the original frame. To further enhance the robustness of this method, we combine pixel scrambling with artificial noise injection. This combined mechanism increases resistance against statistical and visual attacks. Unlike standard encryption algorithms, this hybrid approach is designed to be lightweight, visually effective, and suitable for real-time applications.

Step 5: Performance Evaluation

After encryption and decryption, both the existing and proposed models are evaluated based on quantitative performance metrics. Three primary image quality assessment metrics are used: Peak Signal-to-Noise Ratio (PSNR)**,** Mean Squared Error (MSE)**,** and Structural Similarity Index (SSIM)**.** These metrics are computed between the original and decrypted frames to assess the fidelity and effectiveness of each technique. PSNR gives an indication of the overall distortion, MSE measures the pixel-wise error, and SSIM evaluates perceptual similarity. These values are averaged across multiple frames to ensure consistency and reliability in the results. The comparison table helps in visually identifying which method performs better in terms of visual preservation and structural accuracy.

Step 6: Prediction on New Unseen Test Video

To validate the robustness and generalizability of the developed encryption techniques, a new unseen video is passed through the complete pipeline. The video is first converted into frames, encrypted using both LSB and the proposed scrambling method, then decrypted, and finally reconstructed. The decrypted video frames are again subjected to PSNR, SSIM, and MSE calculations to confirm that the proposed method maintains its performance across different video content. This step simulates real-world deployment, where encryption must work consistently on new data without retraining or reconfiguration. Successful decryption with high similarity scores on new videos demonstrates the applicability and reliability of the proposed method in practical scenarios.

4.2 Video Encryption and Decryption Preprocessing

Video-to-Frame Decomposition

The preprocessing phase in this research begins with the transformation of the raw input video into a sequence of individual frames. This step is essential because both the encryption and decryption techniques used in the study—namely, the Least Significant Bit (LSB) method and the proposed pixel scrambling algorithm—operate on static images rather than continuous video streams. Using OpenCV, the input video is read frame-by-frame and each frame is stored in a dedicated folder with a sequential naming convention (e.g., frame_0001.png, frame_0002.png, etc.). This decomposition simplifies the processing pipeline, allowing for fine-grained control over encryption and decryption at the frame level.

Noise Simulation through Probabilistic Perturbation

To mimic the real-world transmission environment and to improve robustness, controlled noise is introduced during preprocessing using a custom function. This function randomly alters a small percentage of pixel values in each frame by assigning either maximum intensity (255) or minimum intensity (0) to simulate salt-and-pepper noise. This controlled noise is added before the encryption step to evaluate the resilience of each algorithm against minor distortions, a feature often overlooked in traditional studies. It adds an additional challenge to the decryption process and helps in validating the quality of the reconstruction.

Frame Preparation for LSB Encryption

For the existing encryption method, preprocessing involves preparing each noisy frame for LSB manipulation. This is done by applying a bitwise XOR operation with 1, effectively flipping the least significant bit of each pixel value. This bit-level transformation is simple but powerful, enabling data hiding while maintaining a high level of visual fidelity. The preprocessed frames are then saved to a new directory and used for further comparison after decryption. This form of preprocessing is lightweight and suitable for scenarios where computational simplicity and visual integrity are prioritized.

Frame Preparation for Proposed Scrambling Encryption

In the case of the proposed scrambling algorithm, preprocessing entails flattening the noisy image frames into one-dimensional pixel arrays. A fixed seed is used to generate a pseudorandom index list, which is then applied to shuffle the pixel positions, effectively scrambling the spatial arrangement of the image while preserving the pixel values. The fixed seed ensures that the scrambling can be reversed during decryption. This preprocessing step not only ensures strong visual obfuscation but also adds a layer of reproducibility, as the same seed can be used to decrypt and reconstruct the original image without loss.

Organized Storage for Comparative Analysis

To facilitate performance evaluation, the preprocessing stage concludes with the structured storage of all processed frames—both encrypted and decrypted—in separate folders. This structure ensures consistency in file naming and indexing, allowing automated functions to compare original and reconstructed frames using quality metrics like PSNR, MSE, and SSIM. This folder-based organization also supports smooth video reconstruction after decryption, ensuring that decrypted frames can be reassembled into a playable video for visual inspection.

4.3 Model Building

This section outlines how the research builds and implements the encryption and decryption models on the video dataset, using both traditional and novel methods. The goal is to demonstrate how each model is applied to the pre-processed frames, how encrypted videos are reconstructed, and how decrypted outputs are evaluated. The model-building process also includes encryption robustness and the ability to reverse encrypted content with minimal distortion. We examine two models: an existing LSB-based method and a newly proposed scrambling-based encryption approach.

4.3.1 Existing Algorithm – LSB Decryption

Definition and Information

The Least Significant Bit (LSB) algorithm is a widely used method in steganography and lightweight encryption that embeds information into the least significant bits of pixel values. In this study, the LSB algorithm is employed for frame-level video encryption. During encryption, each pixel's LSB is flipped using bitwise XOR operations, subtly altering the image without significant perceptual change. Decryption involves reversing this XOR operation to restore the original frame. The LSB technique is computationally efficient and maintains high visual similarity, making it ideal for environments with limited resources.

How It Works

In this research, the LSB algorithm is applied to each frame extracted from the video. First, random noise is introduced into each frame to simulate a noisy transmission environment. Then, each pixel value is XORed with 1, which flips only the least significant bit of each channel (R, G, B). For decryption, the same XOR operation is performed again, effectively reversing the bit flip and recovering the original values. Since XOR is a symmetric operation, this process ensures that encrypted and decrypted frames match closely—assuming the noise is kept minimal or managed separately. This model is highly efficient and operates at the binary level, but it is sensitive to any modification in the LSB during transmission or compression.

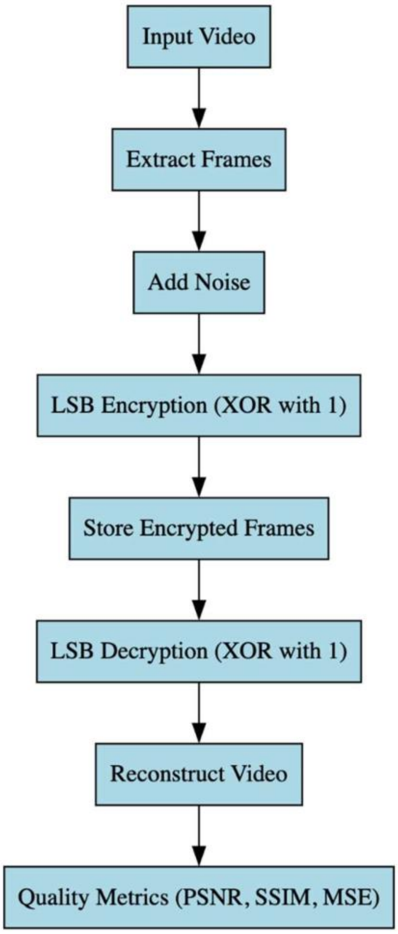
Input Video
Extract Frames
Add Noise
LSB Encryption (XOR with 1)
Store Encrypted Frames
LSB Decryption (XOR with 1)
Reconstruct Video
Quality Metrics (PSNR, SSIM, MSE)

Fig. 4.3.1: Proposed system architecture of LSB.

Algorithm Steps (Architecture)

1. Read each frame from the video.
   Apply simulated noise using a probability function.

1. Perform bitwise XOR with 1 on every pixel (for R, G, B channels).
2. Save the encrypted frame.
3. For decryption, read the encrypted frame.
4. Apply XOR with 1 again to flip the bits back.
5. Save the decrypted frame.
6. Reconstruct video from decrypted frames.

Disadvantages

- **Low Robustness:** Any minor change during transmission (compression, noise) can corrupt the LSB, leading to poor decryption.
- **Weak Security:** As LSB is predictable and reversible, it's vulnerable to statistical and visual attacks.
- **Not Suitable for High-Security Use:** Lacks true cryptographic strength, making it ineffective in highly secure or sensitive environments.
- **Not Resilient to Compression:** Lossy formats like MP4 can alter LSB values.

4.1.2 Proposed Algorithm – Scrambling Encryption

Definition and Information

The proposed algorithm introduces a novel scrambling-based encryption technique that randomizes the pixel positions of each frame using a fixed pseudorandom seed. Unlike LSB, this method does not alter pixel values but instead disarranges their spatial positions, making the image visually unreadable. Scrambling preserves the statistical properties of the image while making it unintelligible to both humans and basic pattern recognition algorithms. Decryption is performed by reapplying the same pseudorandom sequence to restore the original positions of the pixels. The key innovation lies in the deterministic and reversible nature of this shuffling mechanism, combined with noise injection to simulate real-world challenges.

How It Works

Each video frame is first flattened into a one-dimensional array representing its pixel values. A random seed (e.g., 42) is used to generate a unique but repeatable shuffle sequence. This sequence is then applied to rearrange the pixel indices of the flattened frame, effectively scrambling the spatial structure of the image while keeping all pixel values intact. During decryption, the same seed is reused to regenerate the identical index sequence, allowing exact reversal of the scrambling operation. This method is resilient to visual inspection and statistical inference and adds significant complexity for unauthorized reconstruction.

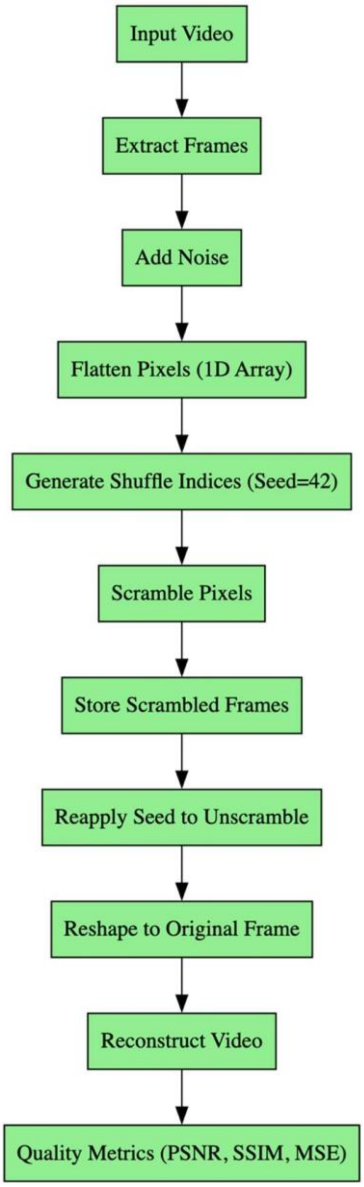
Input Video
Extract Frames
Add Noise
Flatten Pixels (1D Array)
Generate Shuffle Indices (Seed=42)
Scramble Pixels
Store Scrambled Frames
Reapply Seed to Unscramble
Reshape to Original Frame
Reconstruct Video
Quality Metrics (PSNR, SSIM, MSE)

Fig. 4.3.2: Proposed Scrambling Encryption architecture.

Algorithm Steps (Architecture)

1. Read each frame from the video.
2. Apply noise using a fixed probability to simulate perturbation.
3. Flatten the frame to a 1D pixel array.
4. Use a fixed random seed to generate a shuffle index.
5. Rearrange pixels using the shuffled indices.
6. Save the scrambled (encrypted) frame.
7. For decryption, re-generate the same shuffle index using the same seed.
8. Re-map pixels to original positions using the index.
9. Save the decrypted frame and reconstruct video.

Advantages

- **Higher Visual Security:** Scrambled images are entirely unintelligible, unlike LSB- altered images which remain recognizable.
- **Reversible and Deterministic:** Same seed ensures exact decryption without data loss.
- **Robust Against Statistical Attacks:** Since pixel values remain unchanged but locations vary, it defeats histogram-based analysis.
- **No Compression Sensitivity:** Unlike LSB, scrambling works even after format conversion or mild compression.
- **Lightweight:** No complex encryption libraries or high memory overhead needed.