**Advanced Encryption for**

**Quantum-Safe Video Transmission**

**ABSTRACT: -**

This project enables secure video processing, encryption, and watermark embedding, focusing on user authentication, video encryption, and decryption capabilities. Users can register, log in, and upload videos along with watermarks for processing. Using the cryptography library, each uploaded video is encrypted, and its encryption key is split using Shamir's Secret Sharing, ensuring secure key distribution and storage. The encrypted frames are stored separately for later retrieval and decryption. Decryption occurs through reassembling key shares, allowing the original video to be reconstructed, with the watermark extracted from the first frame. The application further provides options to download the decrypted video, view split frames, and explore contact and performance information pages. Employing OpenCV for video processing and secure file handling techniques, this system ensures data confidentiality and integrity through a user-friendly interface and robust back-end encryption mechanisms. The application uses secure upload and storage mechanisms for sensitive data, like key shares and encrypted frames, storing them in predefined folders. Key shares are stored separately, further protecting the decryption process from unauthorized access.

|  |  |
| --- | --- |
| **EXSISTING SYSTEM** | **PROPOSED SYSTEM** |
| The existing system for video transmission primarily relies on traditional encryption techniques, such as symmetric and asymmetric algorithms (e.g., AES, RSA), to secure video data. While these methods effectively ensure confidentiality, integrity, and authenticity, they face significant challenges due to advancing computing capabilities that make them more vulnerable to brute-force and cryptographic attacks. Additionally, the increasing demand for security in digital communication necessitates stronger encryption methods. Many systems currently utilize SSL (Secure Sockets Layer) for secure transmission over the internet, adding a layer of protection; however, this approach may not be sufficient against sophisticated threats. Consequently, the limitations of existing methods highlight the urgent need for more advanced solutions, such as the proposed Hybrid Quantum Video Encryption Framework, to enhance the security of video transmission effectively.  With the rise of more powerful computing technologies, traditional encryption methods face vulnerabilities, as they can be subjected to brute-force attacks and other cryptographic attacks. | The proposed system is a Flask-based web application designed to deliver secure video processing, encryption, and watermark embedding, with a strong focus on user authentication and data confidentiality. This platform allows users to register, log in, and upload video files, along with watermark images, to enhance security and content integrity. Video files are encrypted, and the encryption key is split using Shamir's Secret Sharing technique, ensuring both secure distribution and controlled access to the decryption key. This process of splitting and securing the encryption key guarantees that only authorized users with the required number of key shares can successfully decrypt and reconstruct the video.  Once uploaded, each video undergoes watermark embedding using OpenCV, where the watermark is integrated into each frame. Following this, the watermarked frames are encrypted frame by frame, allowing for individual frame retrieval and flexible management of video data. The encrypted frames are stored separately, making the data accessible only through the reconstruction and decryption processes. This frame-by-frame encryption further ensures that no single frame is exposed in the absence of the full decryption key, enhancing the overall security of the system.  The decryption process is managed through a secure key-sharing mechanism, where users can submit multiple key shares to reconstruct the encryption key. Upon providing enough key shares, the system reassembles the key, decrypts the video, and allows for the extraction of the original watermark from the first frame. This feature is critical for maintaining the integrity of both the original content and the watermark post-decryption. Furthermore, the system enables users to download the reconstructed video, while an additional route provides access to split encrypted frames, allowing users to verify or inspect individual frames if required. |
| **EXISTING ALGORITHM**   * AES,RSA | **PROPOSED ALGORITHM: -**   * Shamir's Secret Sharing and OpenCV |
| The existing system for video transmission security predominantly utilizes traditional encryption techniques, which include a variety of symmetric and asymmetric algorithms such as Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA). These methods play a crucial role in ensuring the confidentiality, integrity, and authenticity of video data during transmission over networks. For instance, symmetric encryption algorithms like AES are favored for their speed and efficiency in encrypting large volumes of data, while asymmetric methods like RSA provide secure key exchange mechanisms.  Despite their effectiveness, traditional encryption techniques face increasing challenges due to advancements in computing power and cryptanalysis techniques. As computational capabilities grow, attackers can employ brute-force methods or sophisticated algorithms to break these encryption schemes, rendering them less secure. This vulnerability is further exacerbated by the rise of quantum computing, which threatens to outpace traditional encryption methods entirely, potentially compromising sensitive data.  Moreover, the current approach often involves using SSL (Secure Sockets Layer) or its successor, TLS (Transport Layer Security), for secure communication over the internet. While SSL/TLS adds a layer of security by encrypting data in transit, it is not immune to attacks, particularly if the underlying encryption methods are weak. Additionally, SSL/TLS alone may not address the complex security needs of video transmission, where high-quality data streams require not only confidentiality but also resilience against interception and manipulation. | **ALGORITHM DEFINITION: -**  The proposed system is a comprehensive and secure web application built with Flask, focusing on video processing, encryption, and watermark embedding. It offers user authentication through secure registration and login, employing hashed passwords for enhanced security. Users can upload videos and watermarks, which are processed using OpenCV to embed the watermark into video frames. Each frame is encrypted with Fernet symmetric encryption, and the encryption key is split using Shamir's Secret Sharing, ensuring secure key distribution and storage. The system allows users to reconstruct the encryption key from shares, enabling the decryption of the original video and extraction of the watermark from the first frame. Additionally, the application features a user-friendly interface for downloading decrypted videos and viewing split frames. It maintains data protection through secure file handling and organizes encrypted and decrypted content systematically. Future enhancements may include an admin dashboard for monitoring user activity, advanced encryption protocols for improved security, and a feedback mechanism for continuous improvement, making the system a robust solution for protecting digital content while facilitating watermarking and encryption. |
| **DRAWBACKS: -**   * Vulnerability to Attacks: Traditional encryption methods are increasingly susceptible to brute-force and cryptographic attacks due to advancing computational power. * Quantum Computing Threat: The rise of quantum computing poses a significant risk to the security of conventional encryption algorithms. * Inadequate Security Measures: SSL/TLS, while providing a layer of security, may not fully protect against sophisticated cyber threats. * Performance Limitations: Traditional encryption algorithms can experience performance bottlenecks when handling large volumes of video data. | **ADVANTAGES: -**   * Comprehensive Library: OpenCV offers a wide range of functions and tools for image and video processing, making it suitable for various computer vision tasks**.** * Real-Time Processing: OpenCV is optimized for real-time applications, enabling fast processing of images and videos, which is crucial for time-sensitive tasks. * Ease of Use: With a user-friendly API and extensive documentation, OpenCV simplifies the development of complex computer vision applications. * Scalable Security: The scheme can easily scale to accommodate any number of participants and shares without compromising security. |

**SYSTEM ARCHITECTURE:**

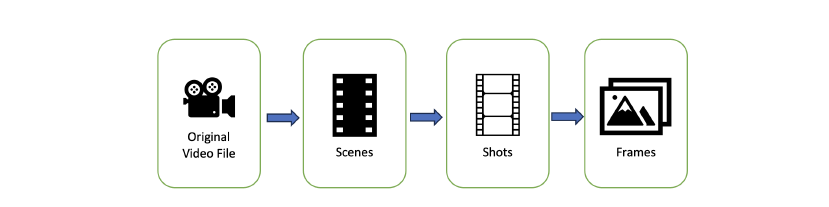


Fig:- proposed model

**MINIMUMSYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* PROCESSOR : Pentium i3 Processor
* RAM : 4GB DD RAM
* HARD DISK : 500 GB

**SOFTWARE REQUIREMENTS**

* BACK END : PYTHON
* OPERATING SYSTEM : WINDOWS 10
* IDE : Spyder3