iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76408

# A Multikey Cryptography Technique for Video Transmission

Hitashree S G[1], Mrs. Sowmya D[2], Harsha S B[3] Aishwarya S[4], Aishwarya S P[5]
*[1, 3, 4, 5]UG students, [2]Assistant Professor, Department of Computer Science and Engineering, JNNCE, Visvesvaraya Technological University, Karnataka, India*

***Abstract: The unauthorized access risk and growing multimedia applications demand secure video transmission over open networks. A hybrid multikey cryptographic technique for secure video transmission without explicit key exchange is presented in this paper. The video is divided into chunks of predetermined size, and each chunk has a unique encryption key that is dynamically created through an ECC-like recurrence-based model. To secure the key-generation parameters, a video identifier (VID), which is derived from the original video, is encrypted using RSA. At the receiver, corresponding keys are created using the decrypted VID and the MAC address that is specific to the device. The confidentiality of each chunk is guaranteed through AES encryption. MSE and PSNR are employed to conduct the experiments and the results show that the encrypted video does not disclose any visual information and the decrypted video has very good visual quality, thus proving the proposed scheme's effectiveness and security.***

***Keywords: Multikey Cryptography, Secure Video Transmission, Hybrid Encryption, RSA, AES, ECC-inspired Key Generation, MAC-based Authentication, PSNR, MSE***

## I. INTRODUCTION

The videos sent over open and distributed networks have become essential in modern applications like online education, video conferencing, surveillance systems, and multimedia streaming. However, the increasing volume of video data and its transmission through unsecured channels make sensitive content vulnerable to eavesdropping, unauthorized access, and data tampering. Traditional video encryption methods often rely on a single static key or simple key exchange processes, which creates significant security risks if the key falls into the wrong hands. Additionally, securely managing and distributing encryption keys is challenging, especially in large-scale and diverse network environments. To tackle these limitations, multikey cryptographic methods have been proposed as an effective way to make video more secure. This paper presents a hybrid multikey cryptography method for secure video transmission. It divides the video into fixed-size parts, encrypting each part with a key that is generated dynamically. The proposed system uses a key generation model inspired by ECC, which creates unique keys based on video content and device-specific parameters. This approach eliminates the need for direct key exchange. A video identifier (VID) from the original video is secured using RSA, while AES-based encryption is applied to individual video chunks to protect confidentiality. On the receiver's side, the same keys are generated independently using the decrypted VID and MAC address, allowing for secure and reliable video reconstruction. Experimental results from standard quality metric evaluation indicate that the proposed method can effectively protect video data while maintaining high visual quality after decryption.

## II. LITERATURE REVIEW

Multimedia applications on open networks have turned secure video transmission into a focused research area. Traditional cryptographic algorithms like DES, AES, and RSA were the main methods used initially for encrypting video data. AES is a symmetric encryption technique that delivers good performance and strong security. However, its reliance on a single static key is a risk because it can lead to key compromise. RSA allows for secure key distribution, but it consumes a lot of power when applied to large video files. As a result, several hybrid encryption schemes have been proposed. These combine the confidentiality of symmetric encryption with the key protection of asymmetric encryption to find a balance between security and efficiency.

It has been explored that multikey and dynamic key generation techniques to improve security for multimedia encryption. Different parts of the video use different keys. This approach limits the impact if one key is compromised. Some studies use chaotic maps or pseudo-random sequences for key generation, while others choose elliptic curve cryptography (ECC) for its strong security features and small key sizes. ECC-based methods are especially attractive for communication systems that need high security. They resist cryptographic attacks while maintaining low computational complexity. However, many existing schemes still depend on explicit key exchange or centralized key distribution, which can increase security risks.

The studies show the need to remove key exchange overhead in any secure video transmission system. Content-dependent key generation methods can create encryption keys based on characteristics of the video or other device-specific factors. These methods allow the receiver to regenerate the encryption keys independently, without needing to send keys directly. This improves security and scalability. This research builds on these ideas to develop a hybrid multikey cryptography technique. It will combine RSA for video identifier protection with an ECC-inspired model that generates keys based on recurrence for dynamic per-chunk encryption. This approach will provide a more secure, efficient, and high-quality video transmission.

## III. PROBLEM STATEMENT

Secure transmission of video data over open networks is a major challenge. This is due to the high sensitivity and large size of multimedia content. Most existing video encryption techniques use a single static encryption key or require explicit key exchange. These methods significantly raise the risk of security breaches if the key is compromised and add extra key management work. While asymmetric cryptography offers secure key distribution, it is not efficient for large video streams. On the other hand, symmetric encryption by itself does not provide strong protection against unauthorized key access. Moreover, many multikey encryption methods rely on centralized key servers or frequent key exchanges. This reliance limits scalability and increases vulnerability to attacks. Therefore, we need a secure, efficient, and scalable video encryption method that can dynamically create multiple encryption keys without needing explicit key exchange, ensures access control at the device level, and maintains video quality after decryption.

## IV. PROPOSED METHODOLOGY

The proposed system uses a hybrid multikey cryptography technique for secure video transmission. It generates multiple encryption keys on the fly and applies them to different video segments without explicitly exchanging keys. The method combines asymmetric and symmetric cryptographic mechanisms with a key generation model inspired by ECC. This approach improves security while keeping efficiency. The entire process includes video preprocessing, dynamic multikey generation, encryption at the sender side, and independent key regeneration and decryption at the receiver side.

### *A. System Overview*

In the proposed approach shown in figure 1, the input video is first divided into fixed-size chunks. This allows for encryption of each segment. A video identifier (VID) comes from the initial portion of the video content and acts as a seed for generating dynamic keys. The VID is encrypted using RSA to protect the key-generation details during transmission. For each video chunk, a unique encryption key is created using an ECC-inspired recurrence relation that takes into account the VID, chunk index, and receiver-specific MAC address. These keys are not stored or sent; instead, they are regenerated independently on the receiver's side using the same parameters. This method removes the need for explicit key exchange.

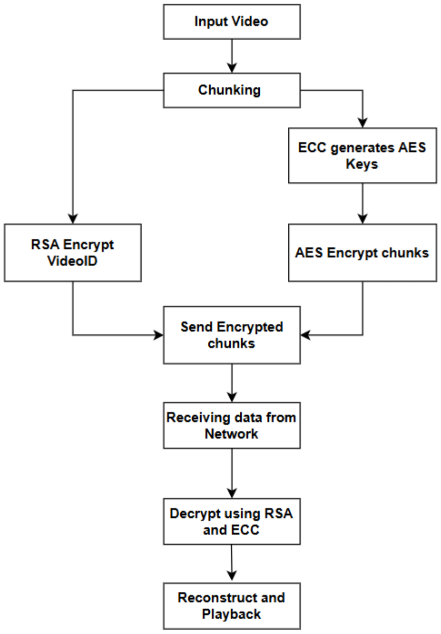


Fig. 1 Multikey Cryptography flow diagram

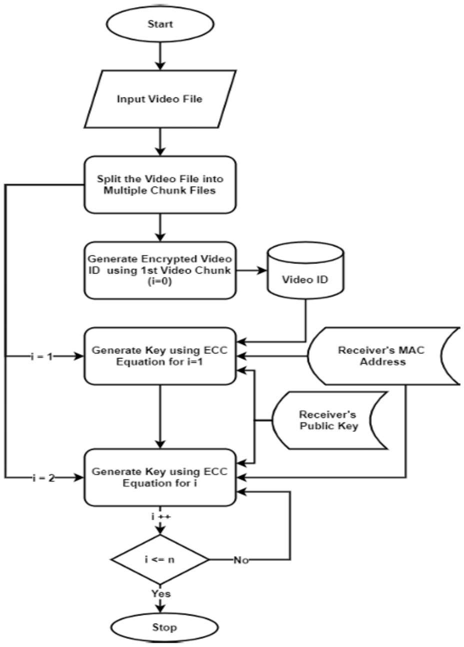

Start
Input Video File
Split the Video File into Multiple Chunk Files
Generate Encrypted Video ID using 1st Video Chunk (i=0)
Video ID
i = 1
Generate Key using ECC Equation for i=1
Receiver's MAC Address
Receiver's Public Key
i = 2
Generate Key using ECC Equation for i
i ++
i <= n
No
Yes
Stop

Fig. 2 Flow diagram of proposed Key Generation technique

### *B. ECC-Inspired Multikey Generation Model*

The multikey generation mechanism is inspired by elliptic curve cryptographic principles and operates over a large prime finite field. Let *x* denote the chunk index, *VID* represent the video identifier derived from the original video, and *Rmac* denote the receiver's MAC address converted into an integer form. The encryption keys are generated using the following recurrence relations as shown in figure 2:

For the first video chunk:

$$Key_1 = x^3 + (VID \cdot x) + Rmac(\mathrm{mod}P)$$

For subsequent chunks:

$$Key_i = x^3 + (Key_{i-1} \cdot x) + Rmac(\mathrm{mod}P)$$

where *P* is a large prime modulus defining the finite field. This recurrence ensures that each video chunk is encrypted with a distinct key, and the compromise of one key does not affect the security of other chunks.

### *C. Video Encryption Process*

At the sender's side, the video file is split into fixed-size chunks. The VID is taken from the first chunk and encrypted with the receiver's RSA public key. Using the ECC-inspired recurrence model, a unique key is created for each video chunk. These keys are then processed with a cryptographic hash function to create fixed-length symmetric keys suitable for AES encryption. Each video chunk is encrypted independently using AES encryption. The encrypted chunks, along with the encrypted VID and necessary metadata, are sent to the receiver. This hybrid encryption method combines the efficiency of symmetric encryption with the secure parameter protection of asymmetric cryptography.

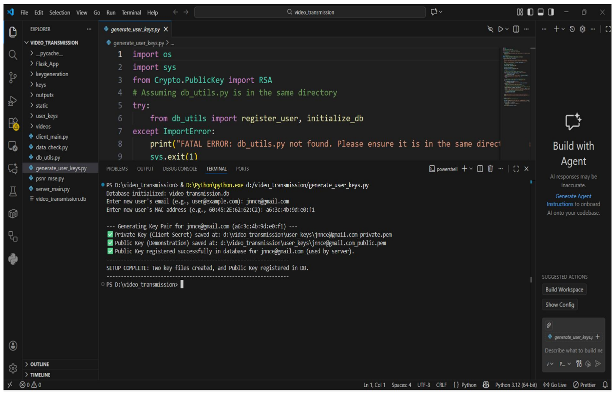
### *D. Video Decryption Process*

At the receiver side, the system first parses the encrypted payload to extract the encrypted VID and video chunks. The receiver uses its RSA private key to recover the VID. With the decrypted VID, the local MAC address, and the known chunk indices, the system regenerates the same multikey sequence using the ECC-inspired recurrence equations. Each regenerated key decrypts its corresponding video chunk. Since the system regenerates the keys independently and does not transmit them, it ensures secure decryption without key exchange. Finally, the decrypted chunks are reassembled to reconstruct the original video.

## V. RESULTS AND ANALYSIS

This section shows the results of using the proposed Multikey Cryptography Technique for Secure Video Transmission. This technique combines ECC-based dynamic key generation, AES chunk-level encryption, RSA metadata protection, and a Client-Server communication framework. RSA keys are created to protect metadata. Authenticated users start the secure transmission session using the user interface. The system monitors every stage of encryption and communication to ensure a smooth workflow.

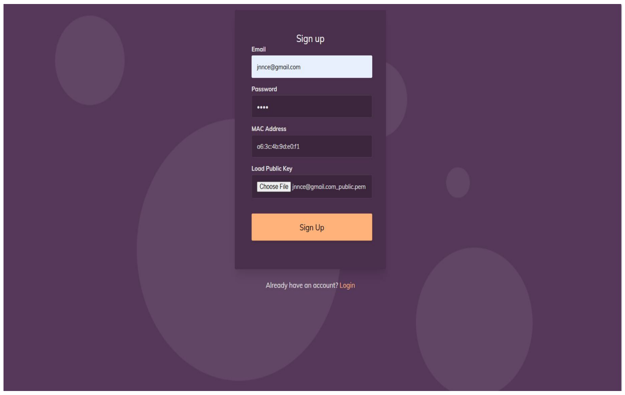
During operation, the Server encrypts each video chunk using ECC-derived AES keys. The Client checks the metadata, regenerates the keys, and decrypts the chunks before reconstructing the final video. Status indicators like "Encrypted," "Decrypted," and "Playback Ready" confirm correct execution and help monitor the system's behavior. The results show that the proposed multikey cryptography model maintains video quality, provides strong security, and works reliably across various test samples.



```python
import os
import sys
from Crypto.PublicKey import RSA
# Assuming db_utils.py is in the same directory
try:
    from db_utils import register_user, initialize_db
except ImportError:
    print("FATAL ERROR: db_utils.py not found. Please ensure it is in the same direct
    sys.exit(1)
```

```
PS D:\video_transmission> & D:\Python\python.exe d:/video_transmission/generate_user_keys.py
Database initialized: video_transmission.db
Enter new user's email (e.g., user@example.com): jnnce@gmail.com
Enter new user's MAC address (e.g., 60:45:2E:62:62:C2): a6:3c:4b:9d:e0:f1

--- Generating Key Pair for jnnce@gmail.com (a6:3c:4b:9d:e0:f1) ---
✅ Private Key (Client Secret) saved at: d:\video_transmission\user_keys\jnnce@gmail.com_private.pem
✅ Public Key (Demonstration) saved at: d:\video_transmission\user_keys\jnnce@gmail.com_public.pem
✅ Public Key registered successfully in database for jnnce@gmail.com (used by server).
------------------------------------------------------------
SETUP COMPLETE: Two key files created, and Public Key registered in DB.
------------------------------------------------------------
PS D:\video_transmission> 
```

Fig. 3 RSA Key Generation Output



Sign up

Email
jnnce@gmail.com

Password
••••

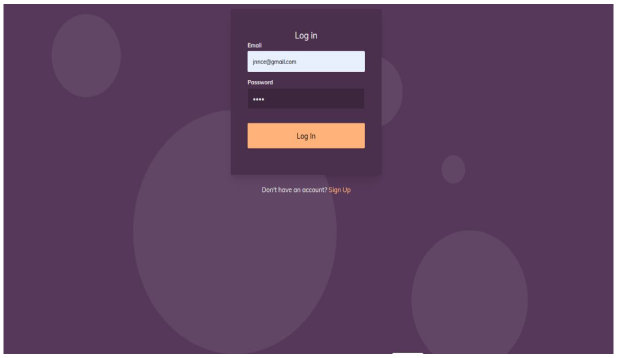
MAC Address
a6:3c:4b:9d:e0:f1

Load Public Key
Choose File jnnce@gmail.com_public.pem

Sign Up

Already have an account? Login

Fig. 4 User Registration Interface

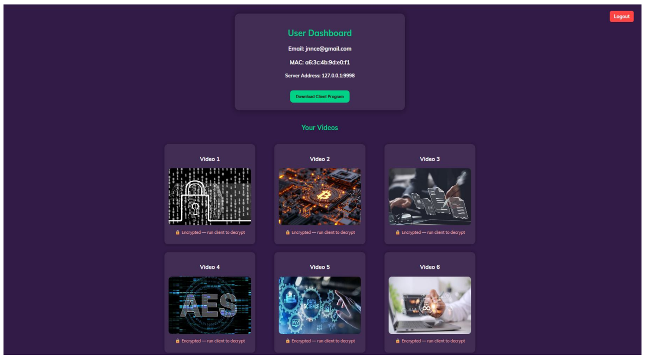


Log in

Email: jnnce@gmail.com

Password: ••••

Fig. 5 Login Interface –Initial Login Page



User Dashboard

Email: jnnce@gmail.com

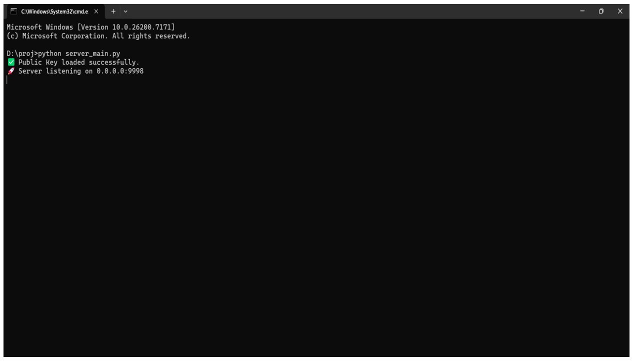
MAC: a6:3c:4b:9d:e0:f1

Server Address: 127.0.0.1:9998

Logout
Download Client Program
Your Videos
Video 1
🔒 Encrypted — run client to decrypt
Video 2
🔒 Encrypted — run client to decrypt
Video 3
🔒 Encrypted — run client to decrypt
Video 4
🔒 Encrypted — run client to decrypt
Video 5
🔒 Encrypted — run client to decrypt
Video 6
🔒 Encrypted — run client to decrypt

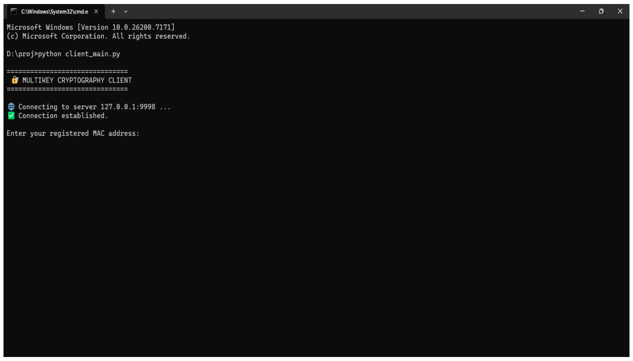
Fig. 6 Login Interface -Logged-In Dashboard

The figure 3 shows the successful generation of RSA public and private keys for the registered user. The private key is stored securely on the client side, while the public key protects the video identifier (VID) during transmission. Figure 4 illustrates the user registration and authentication process. It ensures that only authorized users with valid credentials and registered MAC addresses can access the secure video transmission system. Figure 5 presents the signup form, where new users must enter their email, password, and MAC address to create an account. It shows the interface with user details entered, demonstrating how credentials are provided before registration. This interface ensures that only verified and authorized users can access the secure video transmission system. It forms the first layer of authentication and device-based security binding. Figure 6 shows the interface after a successful login. The user is redirected to the main workspace. This two-step interface demonstrates the system's controlled access mechanism, providing a secure entry point before any cryptographic operations are performed.



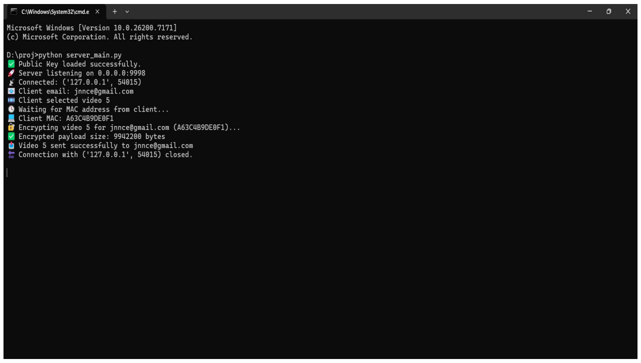
```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

D:\proj>python server_main.py
✅ Public Key loaded successfully.
🚀 Server listening on 0.0.0.0:9998
```

Fig. 7 Server Terminal (Server started)



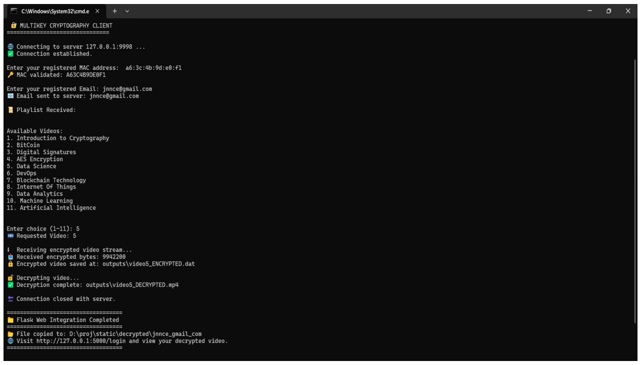
```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

D:\proj>python client_main.py

================================
 🔐 MULTIKEY CRYPTOGRAPHY CLIENT
================================

🌐 Connecting to server 127.0.0.1:9998 ...
✅ Connection established.

Enter your registered MAC address:
```

Fig. 8 Client Terminal (Connected to Server)



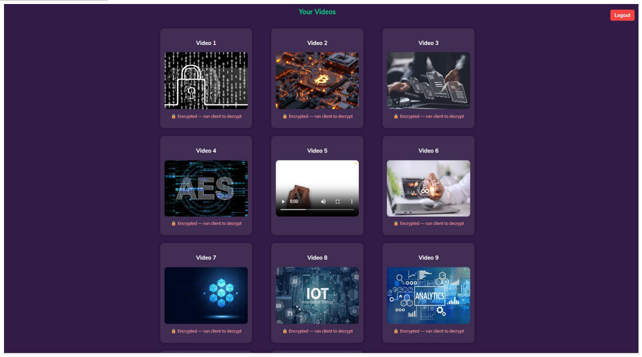
```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

D:\proj>python server_main.py
✅ Public Key loaded successfully.
🚀 Server listening on 0.0.0.0:9998
📡 Connected: ('127.0.0.1', 54015)
📧 Client email: jnnce@gmail.com
🎞 Client selected video 5
🕓 Waiting for MAC address from client...
💻 Client MAC: A63C4B9DE0F1
🔐 Encrypting video 5 for jnnce@gmail.com (A63C4B9DE0F1)...
✅ Encrypted payload size: 9942200 bytes
📤 Video 5 sent successfully to jnnce@gmail.com
🔚 Connection with ('127.0.0.1', 54015) closed.
```

Fig. 9 Encryption Confirmation (Server side)



```
 🔐 MULTIKEY CRYPTOGRAPHY CLIENT
================================

🌐 Connecting to server 127.0.0.1:9998 ...
✅ Connection established.

Enter your registered MAC address:  a6:3c:4b:9d:e0:f1
🔑 MAC validated: A63C4B9DE0F1

Enter your registered Email: jnnce@gmail.com
📧 Email sent to server: jnnce@gmail.com

📜 Playlist Received:


Available Videos:
1. Introduction to Cryptography
2. BitCoin
3. Digital Signatures
4. AES Encryption
5. Data Science
6. DevOps
7. Blockchain Technology
8. Internet Of Things
9. Data Analytics
10. Machine Learning
11. Artificial Intelligence


Enter choice (1-11): 5
🎞 Requested Video: 5

⬇  Receiving encrypted video stream...
📥 Received encrypted bytes: 9942200
🔒 Encrypted video saved at: outputs\video5_ENCRYPTED.dat

🔓 Decrypting video...
✅ Decryption complete: outputs\video5_DECRYPTED.mp4

🔚 Connection closed with server.

===================================
📁 Flask Web Integration Completed
===================================
📂 File copied to: D:\proj\static\decrypted\jnnce_gmail_com
🌐 Visit http://127.0.0.1:5000/login and view your decrypted video.
===================================
```

Fig. 10 Decryption Confirmation (Client side)



🔒 Encrypted — run client to decrypt

Fig. 11 Decrypted Video Displayed on Dashboard

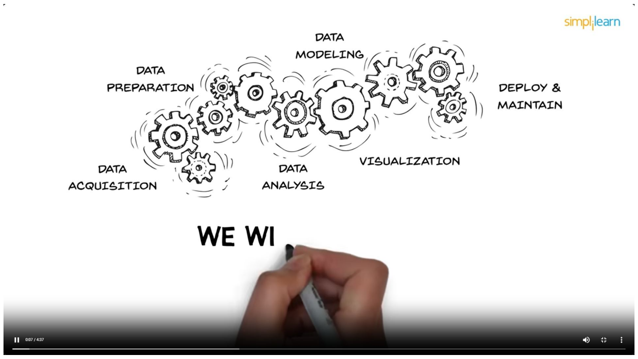


Fig. 12 Decrypted Video Playback on UI

Figure 7 shows the Server program running in a command prompt window and actively listening for client connections. It performs video chunking, ECC-based key generation, AES encryption, and RSA metadata protection. The terminal confirms that the Server is initialized and ready for secure communication. Figure 8 shows the Client program connected to the Server and ready for secure communication. The Client uploads videos, receives encrypted chunks and RSA-secured metadata, verifies integrity, regenerates AES keys using ECC, and decrypts the video chunks. The terminal confirms the connection is successful and the workflow is ready. Figure 9 shows the Server completing the encryption of the uploaded video. Each chunk is encrypted using ECC-derived AES keys, and its metadata is secured using RSA. The terminal output confirms that all encryption steps finished successfully and the Server is ready to transmit the encrypted data to the Client. Figure 10 shows the Client completing the decryption of all received video chunks. The Client regenerates AES keys locally using ECC and decrypts each chunk correctly. The terminal confirms successful decryption and proper reconstruction, which validates the reliability of the multikey decryption process. Figures 11 and 12 show the final stage of secure video transmission. Figure 11 displays the decrypted video listed on the Client dashboard after successful reconstruction, while Figure 12 shows the video playing in the Client UI. These screenshots confirm successful decryption and full recovery of the original video without quality loss. The performance of the proposed multikey cryptography technique was evaluated using Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) metrics, as shown in table 1. For encrypted video data, the MSE values were significantly high, while the PSNR values were low. This indicates strong visual distortion and effective concealment of video content. It confirms that the encryption process successfully prevents unauthorized visual interpretation. In contrast, the decrypted video showed very low MSE values and high PSNR values, demonstrating that the original video content is accurately reconstructed on the receiver side. The results show that the ECC-inspired multikey generation and AES-based encryption and decryption process achieve strong security without compromising video quality.

TABLE 1
MSE and PSNR Values for Encrypted and Decrypted Video Frames

| Sample No. | Video No. | MSE (Original vs Encrypted) | PSNR (Encrypted) | Average MSE (Decrypted) | Average PSNR (Decrypted) |
|---|---|---|---|---|---|
| Sample 1 | Video 8 | 11619.9570 | 7.4788 dB | 0.083335 | 58.922523 dB |
| Sample 2 | Video 5 | 16775.0898 | 5.8842 dB | 0.083330 | 58.922794 dB |
| Sample 3 | Video 3 | 18691.1426 | 5.4144 dB | 0.083339 | 58.922306 dB |
| Sample 4 | Video 9 | 18598.2930 | 5.4361 dB | 0.083339 | 58.922331 dB |
| Sample 5 | Video 11 | 13340.9834 | 6.8789 dB | 0.083324 | 58.923092 dB |

## VI. Conclusion and Future Scope

This paper presents a hybrid multikey cryptography technique for secure video transmission. It removes the need for explicit key exchange while ensuring strong confidentiality and high reconstruction quality. The video is divided into fixed-size chunks, and each chunk is encrypted using dynamically generated keys from an ECC-inspired recurrence-based model. This system reduces the risk of a single key being compromised. Using RSA to protect video identifiers and AES for chunk-level encryption strikes a good balance between security and efficiency. Experimental results based on PSNR and MSE metrics show that the encrypted video has noticeable visual distortion, while the decrypted video gets close to lossless reconstruction. This validates the correctness and effectiveness of the proposed method.

As future work, the proposed framework can be expanded to support real-time video streaming and adjustable chunk sizing to improve performance in different network conditions. The system can also be made more efficient for limited-resource environments, such as mobile and IoT devices, by using lightweight cryptographic methods. Additionally, formal security analysis and testing against complex cryptographic attacks can be examined to enhance the strength of the proposed multikey cryptography scheme.

## REFERENCES

[1] Youcef Fouzar, Ahmed Lakhssassi and M. Ramakrishna, “A Novel Hybrid Multikey Cryptography Technique for Video Communication”, IEEE Access, 21 February 2023

[2] M. A. Al-Shabi, “A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security”, International Journal of Scientific and Research Publications, Volume 9, Issue 3, March 2019

[3] Nitin Jirwan, Ajay Singh, Dr. Sandip Vijay, “Review and Analysis of Cryptography Techniques”, International Journal of Scientific & Engineering Research Volume 4, Issue3, March-2013

[4] Abid Murtaza, Syed Jahanzeb Hussain Pirzada, Liu Jianwei, “A New Symmetric Key Encryption Algorithm With Higher Performance”, International Conference on Computing, Mathematics and Engineering Technologies, iCoMET 2019

[5] Mitali, Vijay Kumar and Arvind Sharma, “A Survey on Various Cryptography Techniques”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 4, July-August 2014

[6] Raza Imam and Faisal Anwer , Qazi Mohammad Areeb, Abdul Rahman Alturki and Faisal Anwer, “Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status”, IEEE Access, November 30, 2021.

[7] Suman Kalyan Ghosh, Sachin Rana, Anushikha Pansari, Joydev Hazra, Satarupa Biswas, “Hybrid Cryptography Algorithm For Secure And Low Cost Communication”, International conference on Computer Science,Engineering and Applications, March 2020

[8] Md. Atiullah Khan, Kailash Kr. Mishra, N. Santhi, J.Jayakumari, “A New Hybrid Technique for Data Encryption”,Global Conference of Communication Technologies(GCCT),2015

[9] Lili Yu, Zhijuan Wang, Weifeng Wang, “The Application of Hybrid Encryption Algorithm in Software Security”, Fourth International Conference on Computational Intelligence and Communication Networks,2012

[10] Neal Koblitz, Alfred Menezes, “The State of Elliptic Curve Cryptography”, Designs, Codes and Cryptography, 19, 173–193, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands,2000

[11] Mohammad Ayoub Khan, Mohammad Tabrez Quasim, Norah Saleh Alghamdi and Mohammad Yahiya Khan, “A Secure Framework for Authentication and Encryption Using Improved ECC for IoT-Based Medical Sensor Data”, IEEE Access, March 25, 2020

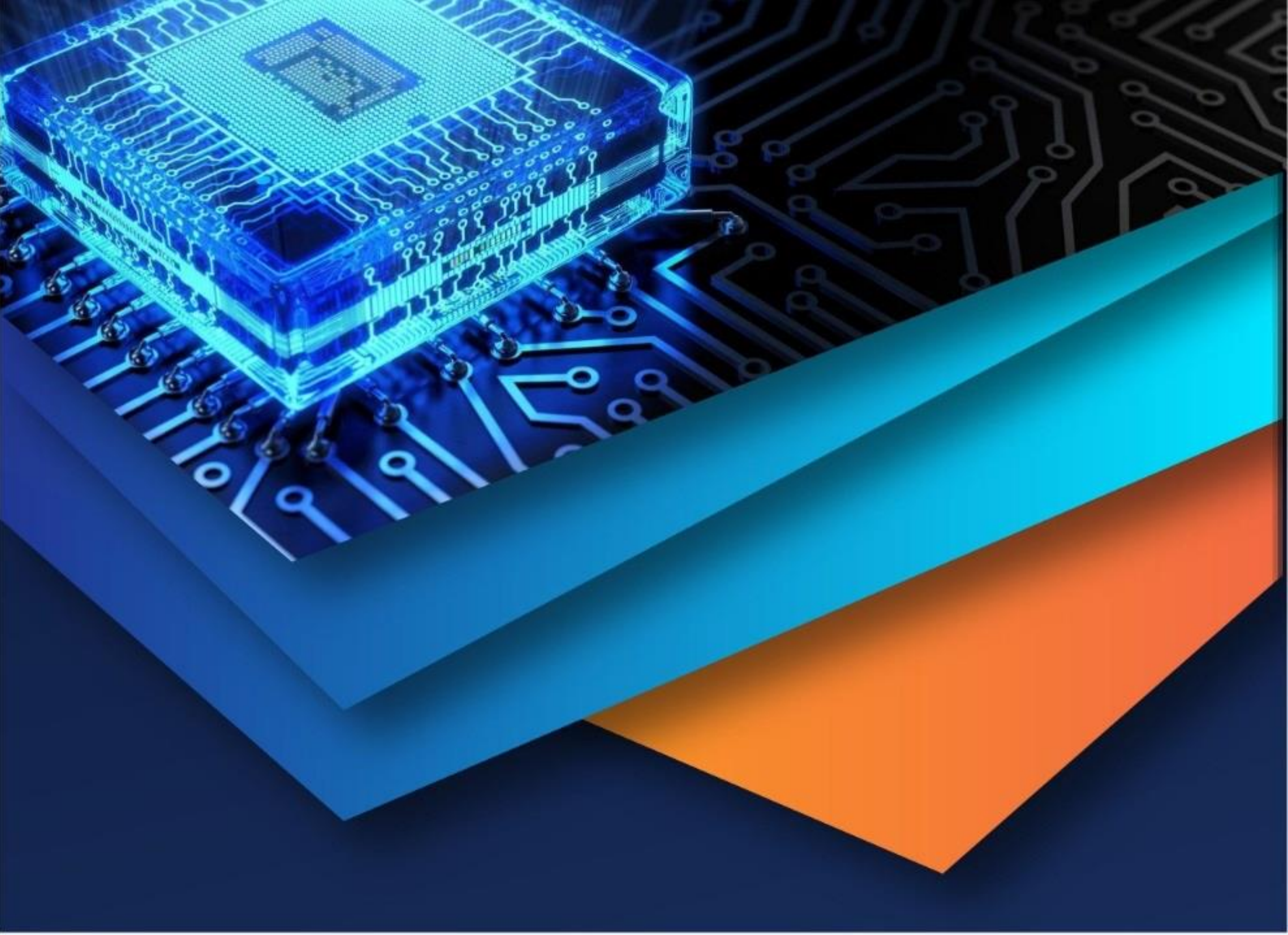
[12] M. Abomhara, Omar Zakaria, Othman O. Khalifa, “An Overview of Video Encryption Techniques”, International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010

[13] Nilanjan Sen, Ram Dantu, Jagannadh Vempati, Mark Thompson, “Performance Analysis of Elliptic Curves for Real time Video Encryption”, National Cyber Summit Research Track,2018

[14] Lo’ai Tawalbeh, Moad Mowafi, Walid Aljoby, “Use of elliptic curve cryptography for multimedia encryption”, IET Information Security, 2013, Vol. 7, Iss. 2, pp. 67–74, November 2012

[15] Sanjeev Kumar, Madhu Sharma Gaur, Prem Sagar Sharma, Deepkiran Munjal, “A Novel Approach of Symmetric Key Cryptography”, 2nd International Conference on Intelligent Engineering and Management (ICIEM),2021

[16] Jhansi Menda, Bhargav Naidu Marrapu, Ravi Teja Bheri, Chandra Sekhar Naidu Chebodula, “Efficient and Secure Video Streaming Using Multi-Key Cryptography”, International Research Journal of Modernization in Engineering Technology and Science, Volume:06, Issue 11, November-2024



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY