Journal of Applied Sciences, Information and Computing

Volume 5, Issue 2, November 2024

School of Mathematics and Computing, Kampala International University

ISSN: 3007-8903 https://doi.org/10.59568/JASIC-2024-5-2-06

A secured channel for transmitting multimedia contents in peer-assisted networks

[1]O. E. Ojo, [2]M. K. Kareem, O. A. [3]Ayilara-Adewale, [4]C. N. Amahia

[1]Department of Information Technology, Osun State University, Osogbo, Nigeria.
[2]Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria.

[1]oluwafolake.ojo@uniosun.edu.ng, [2]kareemmk@funaab.edu.ng, [3]oluwatobi.ayilara-adewale@uniosun.edu.ng, [4]cnmeister17@gmail.com

Abstract:

The Internet's explosive growth has raised demand for effective online video streaming services and cast doubt on the viability of peer-to-peer (P2P) networks and conventional content distribution networks (CDN). To facilitate efficient video streaming, Peer-Assisted Networks (PA-CDNs) integrate the best aspects of P2P networks and CDNs. Transferring data content across a protected channel platform is crucial since the PA-CDNs strategy uses both servers and peers, optimizing both delivery methods to offer the best possible user experience. However, it is vulnerable to assaults. By guaranteeing secure access management and addressing security concerns, this study offers a strong security model to strengthen and improve security in PA-CDN network environments. The outcome of the experiment indicates that the strategy works. Strong security model to strengthen and improve security in PA-CDN network environments. The outcome of the experiment indicates that the strategy works.

Keywords: Content Delivery Networks, Peer-Assisted, Multimedia Applications, Network crusty.

1. Introduction

The recent enhancement of multimedia on the internet has significantly altered the Internet usage pattern due to multiple demands for multimedia applications like online video streaming, video conferencing, webinars, and presence on social media. This increase is mainly due to high customer demand, virtual work, online classes, and entertainment through popular social media like YouTube, Netflix, Amazon Prime, Facebook, Instagram and many more. These platforms,

Characterized as mainstream Internet applications, have experienced fast and steady growth in real-time video transmission (Cheng et al., 2019; Jack et al., 2021; Duan et al., 2020). With the high demand for video transmission, the streaming service provider needs to ensure that they meet the demands of the users at every point in time. There are two categories of video streaming, they are: live video streaming and on-demand video streaming. Live video allows real-time dissemination of contents and synchronization of video playback (Sadek *et al*. 2015). In contrast, on-demand

video streaming has the advantage of allowing users to watch any video clips anytime, but there is no synchronization of the same video clips with different users (Majeed et al., 2017). The content delivery network (CDN) is a client-server service network that enables it to meet the demands of video streaming. It is a distributed server system that enables the delivery of videos and web content to the geographical location of the user (Peng, 2004). CDN is a two-layer client-to-server network that replicates the content of the traditional server to different CDN servers; the clients get resources from the nearest CDN server. The video content could be pre-recorded or live videos (Lee et al., 2020). It stores a cached version of its content in multiple geographical locations. In term of service delivery, it speeds effective in delivering web content with high traffic and globally reached websites, the speed of the content is delivered to the users is based on the closeness of the CDN server to the user's geographical location (Yan et al. 2019; Sarddar et al., 2017). There are many benefits of using CDN and they include reduced server load, high speed of content delivery, rise in concurrent users on a server (Lin, 2020). However, the disadvantages of CDN are cost-effective, complex, CDN could be blocked by network filters which could hinder content from being uploaded, and also Geo-location could be a barrier (Patel, 2018). Research showed that CDNs are not sufficient because they strained when there are high demands by the video users, therefore, there is a need for another content delivery technique that can address the drawbacks of CDNs. Another Content delivery method is known as Peer-Assisted network (PA-CDNs) was introduced to address the inadequacies in CDNs. PA-CDNs combines the techniques of CDNs, and Peer-to-Peer (P2P) networks for content delivery (Nacakl, & Tekalp, 2020). PA-CDN is a hybrid architecture that combines the advantages of both Content Delivery Networks CDNs and P2P systems (Rodrigues *et al*., 2021). The basic design idea of P2P systems is to enable users to act as both clients and servers, it allows the upload and download of content in the network, uploading bandwidth of end-users is efficiently utilized (Siano et al., 2019). Despite the advantage of PA-CDNs over CDNs, PA-CDNs are still vulnerable to attacks (Anjum *et al*., 2017). Furthermore, the stress on the network infrastructure has increased with the need for better and more secure models to accommodate the increased traffic on peer-assisted networks.

Cryptography is a method for securing information communication, it uses mathematical methods with algorithms for transforming information in ways that are difficult to decipher while Steganography is a technique of embedding any form of data (such as audio, image, or video) inside another data (Santiago, 2019). The steganography method is also used to embed messages or information within computer files, electronic communications which may include steganography coding inside of a transport layer (Khare et al, 2011; Singh, 2017; Rajamani et al, 2020).

This study proposes a secured medium suitable for transmitting multimedia contents such as video files across the Internet. The proposed scheme adopts the steganography and cryptography techniques to improve user's quality of experience with Internet-based video transmission within peer-assisted networks. This scheme introduces the concept of adding a secret file to cover video in the steganography phase such that, during transmission, the original video contents are hidden in the secret files which makes the content appear as a file as against the convention methods presented in form of video images or message bit. Additionally, the proposed security model is designed to be suitable for securing video data within PA-CDN network environments because the private keys are not included in the cover images. The private keys are transmitted directly to the authorized peers or users via SMS on mobile phones to avoid the free-riding peers or unwanted peers accessing the contents.

2. Literature Review

Abbas et al. (2020) combined the strength of symmetric and asymmetric encryption techniques to ensure that sensitive cloud data is reliably secured during transmission. A hybrid encryption was implemented using the Advanced Encryption Standard (AES) and River-Shamir Adleman (RSA) algorithms. The encrypted data was embedded in an image using the least significant bit (LSB) algorithm, and SHA hashing was used to validate the information received reliably. The result showed that the method achieved secured key management and rapid data encryption, instilling confidence in its reliability.

Abikoye et al. (2020) developed a hybrid system using Twofish and Triple data encryption (3DES) of the cryptography algorithm with the least significant bits of Steganography to resolve the challenges of hacking and attacking biometrics. Hough transform Daugman rubber-sheet model and log Gabor filter were adopted for the iris image normalisation segmentation and feature extraction, generating an iris template. This Iris template was then encrypted using 3DES and Twofish algorithms to get an unreadable format (Cipher image). The cipher image was embedded into a cover image to get a stego-image using LSB. The study achieved two layers of security method: embedded stego images which can resist attack from hackers.

Verma (2021) proposed a hybrid technique using cryptography and steganography algorithms. The study combined the least significant bit algorithm with the Affine Cipher encryption technique to create a secure channel for sending information, making it less visible. The results showed that the algorithms performed better than other algorithms of a similar approach because integrity and confidentiality of data were achieved, but distortion in the stego-image was noticed.

 https://doi.org/10.59568/JASIC-2024-5-2-06

Salama et al. (2023) presented a multimodal cancelable biometric system method that adopts steganography and cryptography techniques using three parameters: voiceprints, fingerprints, and facial images. The key features of the voiceprint were verified using Mel frequency cepstral coefficients (MFCCs), and Steganography was used to secure the key extracted features from the voiceprints by using block-based singular value decomposition (BSVD) to embed them in a facial image. The double random phase encoding encryption algorithm generated the final cancelable templates. The experimental result showed that the developed system performs highly as a verification system using MFCCs and different transforms.

Raj and Maheswaran (2023) the study proposed a system that combines image steganography and a cryptography approach to enhance secured file sharing. Image steganography was used to embed the secret file inside an image cover file, and then a cryptography algorithm was applied to encrypt the image cover file. The encrypted image cover file can be transferred securely between the sender and receiver. To decrypt the file, the receiver needs to decrypt the encrypted image cover file first and thereafter extract the secret file using the steganography approach. The result showed that the system gives highly secured channel for file sharing by combining the strength of steganography and cryptography.

Rajabi-Ghaleh et al. (2024) introduced a sweeping computational ghost imaging (SCGI) encryption system that combines steganography and cryptography to achieve high speed and improved data security. SCGI needs a few pictures for image reconstruction, which gives a faster image representation and transmission; then encryption is done using steganography and cryptography, which offers a stego image. The stego image is sent to the receiver for decryption. The result showed that combining steganography and cryptography encryption by SCGI has sturdy security with an increasing eavesdropping percentage, and it also increases encryption by up to 90% in relation to the traditional encryption method.

Oduguwa and Arabo (2024) presented a hybrid passwordless authentication system using cryptography, steganography, physiological, and behavioural biometrics. The technique used threat modelling to identify and assess possible threats to determine the pertinent security requirements. Then, it selected the appropriate cryptographic techniques and used comparative analysis to evaluate the usability impacts and security in relation to the orthodox password-based systems. The study achieved a reduced authentication timeframe within 2 seconds compared to traditional password systems and eliminated the need to store passwords and their hashes.

Zhang et al. (2024) proposed public key steganography based on elliptic curve cryptography and a generative model to enhance high security with quite small key sizes, which will help environments with inadequate resources. The study discusses the drawbacks of the existing steganography especially the difficulties with key updating, key agreement and user expansion. The experimental result showed that the techniques adopted were able to provide a secure and efficient key exchange process without requiring preshared keys which addressed the significant challenge in the conventional approach.

Abd Zaid et al. (2024) applied Rivest cipher 4 (RC4) encryption algorithm to convert plaintext into ciphertext, after which the least significant bit (LSB) is used to embed it into an image. The experimental results showed that emerging both techniques added extra security to the traditional steganography and also the stego image retained its' quality.

Ashari et al. (2024) proposed the use of LSB steganography on images adopting the triple data encryption standard (3DES) with message-Digest Algorithm 5 (MD5) hash for text files. The study concludes that the higher the messages embedded in the image, the more obvious the pixel variance becomes. It also tested robustness in the context of rotation and attacks, which showed that attacks can stop message extraction.

The literature reviewed shows that the fusion of stenography and cryptography has achieved a certain level of security in this field. Nonetheless, the key limitations inferred from the existing systems are:

a) **Inadequate security in decryption key:** The traditional techniques used in the existing literature embed the decryption key in the cover image. For example in the model presented by (Rajabi-Ghaleh et al., 2024) and (Raj and Maheswaran, 2023) It was also shown that the decryption key is hidden in the cover image. This approach is not totally secure because intruders can easily access the key for decryption.
b) **Stealth and Concealment:** The principal aim of steganography is to conceal the existence of hidden data. The approach adopted by Verma (2021) and Ashari *et al.* (2024) showed that after achieving the integrity and confidentiality of the data transmitted, distortion and variation in the stego image were observed. When providing a secured channel for transmission information, it is essential to retain the quality of the stego image because if there is a prominent reduction in the image quality, it can stir suspicion with the potential of exposing the concealed message.

In addressing the challenges raised above, this study proposed a robust fusion security model that can secure the transmission of video content while maintaining quality. In addition, this technique proposed another approach to transmitting the decryption keys to the authorized message receivers to prevent hackers or unwanted personnel.

 https://doi.org/10.59568/JASIC-2024-5-2-06

3. Methodology

This section discusses the proposed security system (called ESecureVideo) which provides a two-step embedding and encryption model to send or transfer files within decoy files, to mask the secret message from malicious entities, thus, reducing the risk of information interception and/or compromise and maintaining data integrity. ESecureVideo is a rare combination of the least significant bit technique (LSB) and advanced encryption standard (AES) is used in securing communications. The system involves concealing (embedding) a file inside a video and encrypted with a key. The encoded information can only be decrypted, thereafter de-embedded by the intended recipient who will receive the key from the network provider via short message service over a secure automated simplex channel. The system is broken into several stages. The first stage is where the secret file would be imported into the system; in this case, the file can be a text file, image, or video. The second stage is where we import the cover video; the cover video is the video we would be embedding a file into. The third stage allows the user to use an automatic randomly generated string key to encrypt the files using AES Algorithm and embed the secret file inside the cover video to make an encrypted-stego-file which represents the final processed file.
The ESecureVideo Architecture as shown in Figure 1 consists of the User module, Stego phase, Encryption phase, and Key transfer module. The user module presents a graphical user interface (GUI) at the application layer of the network from the sender-end. In this module, the sender selects the original video content to be transmitted. Additionally, the features in the module also allow the sender to select the preferred secret File. At the stego phase, the selected video content and secret are received from the user module. Thereafter, the stenographic encoder hides the original video content in the secret file using the standard LSB; it hides the video frames of the video at strategic positions in the secret message. The output from the stego phase is the stego-file as seen in Figure 1. The stego-file serves as input into the encryption phase. At the encryption module, a private key is generated using the AES algorithm; the key is used for encryption as well as decryption. Another main activity in this phase is the encryption of the stego-file.

The output from the encryption module is encrypted-stego-file. The encrypted-stego-file is then transmitted to the recipients via the Internet. The main function of the key transfer module is to send the generated private key to the authorized recipients through mobile phones. At the receiver end, the encrypted-stego-file is decrypted using the received decryption key to produce a stego-file. The content of the stego-file is de-embedded using the stenographic decoder which separates the video content from the secret file.

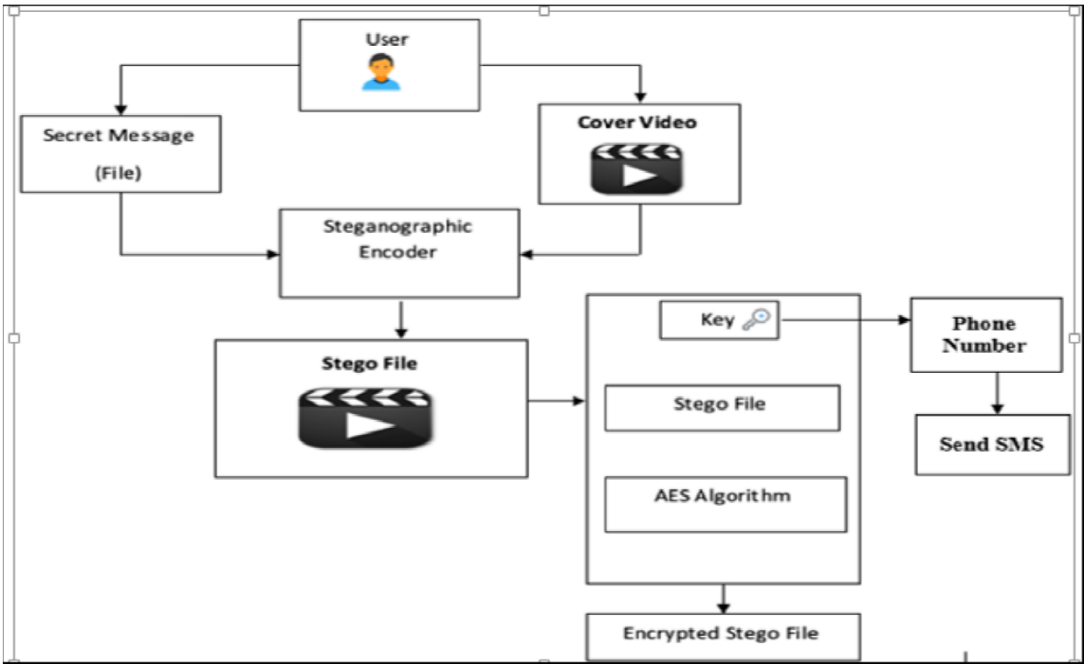

Fig. 1 ESecure Video Architecture

https://doi.org/10.59568/JASIC-2024-5-2-06

The operational definition of the components in the ESecureVideo architecture is given below: **User**: This is a person who uses or utilizes the network service.

Cover Video: The cover video is also known as a decoy video. The user embeds the secret file into a decoy video. The decoy video is what is transferred over the network, and to the ordinary man, the video is just what it looks like a video. However, only the intended recipient of the embedded message is privy to the fact that there is an embedded file within; the recipient will also possess the defined key with which to decrypt and de-embed the secret file.

Secret Message (Hidden File): This is the file to be embedded in the cover video. Information seems to have lost its value. It is sent about carelessly. Phone lines can be tapped, emails hacked, and e-mails intercepted. At some point in time, one will need to give someone a message and there can be no chance of a third party getting their hands on it. This system disguises any hidden file so that it cannot be compromised or intercepted.

Stego File: This is the generic name given to a completed steganography file; it is the method of embedding multimedia data such as text, image, animation, or video within another data.

Key: This is the user-defined password that would be needed to decrypt the file. This key is derived via an automatic random generation. To encrypt a file, we need a public key.

AES: Advanced Encryption Standard (AES) Algorithm is an encryption and decryption algorithm.

Encrypted Stego File: This is the completed encrypted stego (decoy) file, which contains the embedded (hidden) file. The stego-file could be a file document (folder) or video; this can be sent to the intended recipient via any acceptable means such as e-mail and so on.

Steganography Process: This process involves embedding secret information such as video, audio, text file in a manner that will make it difficult for intruders to detect. A high capacity data embedding approach using the LSB insertion technique is used where the aim is to hide information in specific frames of the video and specific positions of the frame using LSB substitution.

Algorithm 1: ESecureVideo

Input: Original Video (*V*), Secret File (*F*)
Procedure:
1. **START**
2. **For** any sendPeers $\in$ Users (*N*)
3. Select $F_1, F_2, \ldots \ldots F_n$
4. Import $V_1, V_2, \ldots \ldots V_n$
5. **Begin** stego-process
6. {
7. **If**(F_n && $V_n \neq false$)
8. {
9. Activate stego-encoder
10. Embed V_n in F_n, $such\ that, \nexists V_n$ in the cover image.
11. }
12. return stego-file(*S*)
13. }
14. end stego-process
15. **Begin** encryption-process
16. {
17. **If**($S_n == true$)
18. {
19. Generate private key(PK) and goto step 28
20. Encrypt S_n
21. }
22. return encrypted-stego-file(E_n)
23. }
24. end encryption-process
25. }
26. Transmit $E_n \leftrightarrow$ recieverPeers $\in N$
27. }
28. **Begin** key-transfer
29. {
30. **If**($PK == true$)
31. {
32. Check if(recieverPeers $\in N$)
33. Accept Phone number (P_n)
34. Send $PK \leftrightarrow P_n$
35. }
36. Else goto Step 19
37. **Begin** decryption-process
38. {
39. **If**(PK && $E_n = true$)
40. {
41. decrypt E_n, such that S_n is revealed
42. de-embed S_n, such that F_n and V_n is separated
43. }
44. return V_n
45. }
46. end decryption-process
47. **STOP**

Output: E_n, V_n

https://doi.org/10.59568/JASIC-2024-5-2-06

The steps involved in the ESecureVideo are summarized in Algorithm 1, the inputs into the system are original video and secret files depending on the sender selection (represented as sendPeers in the Algorithm). The expected output from the system is the encrypted-stego-file which is received by the recipients (represented as receiverPeers in Algorithm 1). The activity in the user module is represented in Lines 2-4 of Algorithm 1. The representations of all the features in the stego phase are presented in Lines 5 -14 of the Algorithm; the end product from the stego phase is stego-file which is represented as "return stego-file" in Line 12 of Algorithm 1. Similarly, all the processes involved in the encryption phase are translated to pseudo-code as shown in Lines 15-24 of Algorithm 1. The key transfer module is represented in Lines 28 -36 of the Algorithm. The encryption phase determines whether the key transfer is activated or not; because the key transfer module depends solely on key generation. Therefore, there is a link between the encryption phase and key transfer module as shown in Lines 18 and 36 of the Algorithm. Lastly, the decryption process is represented in Lines 37 -46 of Algorithm 1.

4. Implementation

The ESecureVideo algorithm presented in Section 3 is implemented using Java programming language; the viability from the implementation with Java brings to life the development of the system, employing the use of steganography to embed a secret video inside a decoy file, then the stego file is further encrypted. The output design of the system is illustrated consequently and the performance evaluation. Specifically, this section presents the ESecureVideo prototype. Upon launching the software, the first thing that appears is the application home screen as shown in Figure 2. Herein, the user would decide specifically what to do, either to embed or de-embed a file. The embed process is to create a steganography file while the de-embed process reveals the hidden steganography file.

Furthermore, the encryption and decryption methods are used to secure the file before it is sent to the recipient. The Embed File Screen is shown in Figure 3; the user clicks on the EMBED button and the system redirects the user to a window where the user can select the video and the file to embed in the video before clicking on the EMBED button. Once the embedding process is completed, a Stego file containing the embedded file is automatically created on the user's desktop, from whence the file can be accessed. Subsequent processes, such as encryption and decryption are also saved to that directory.

In addition, the encryption and decryption menu is shown in Figure 4, this is the main interface of the encryption screen. The user would be able to select either option encrypt or decrypt. To encrypt a file, we need a private key; this key is a user-defined password that would be needed to decrypt the file. The key generation menu is depicted in Figure 5; this key is derived via an automatic random generation using the AES mechanism. After the key is generated, the user enters the recipient's cell phone number where the private key will be sent; the user must have internet access.

Figure 6 depicts the menu for decrypting the encrypted-stego- file using the private key. Figure 7 shows the field where the decryption key is requested. The recipient simply enters the key received from the sender. The decryption process will show complete and both the embedded file and the decoy file can be accessed on the user's computer. Figure 8 shows where the file(s) can be encrypted and decrypted. After the decryption, the file directory will contain the embedded file(s), the de-embedded file(s), the encrypted file(s), and the decrypted file(s). The process uses different video sizes and the results show that the ESecure Video is efficient for securing information over PA-CDNs.

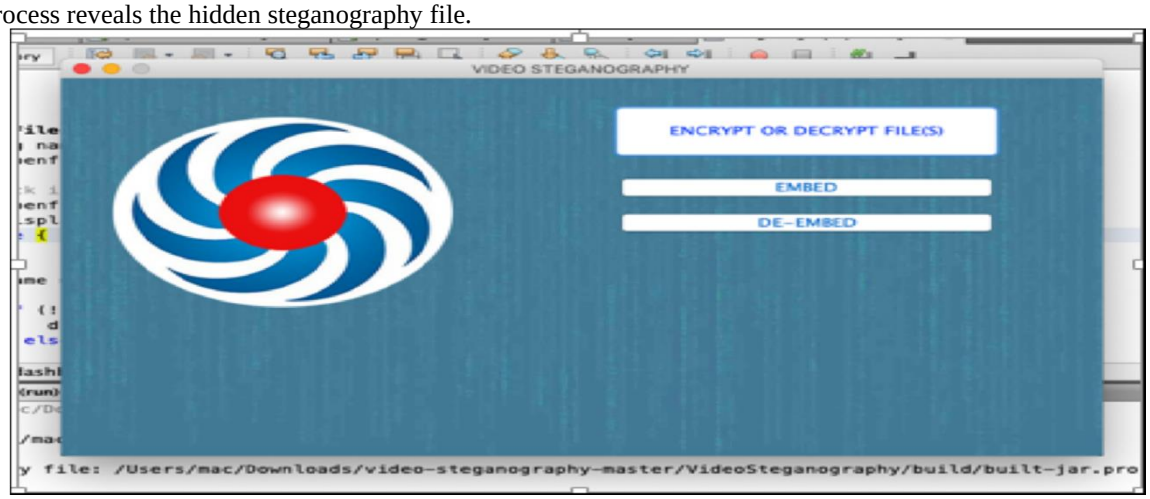

Fig. 2 The Home Screen

https://doi.org/10.59568/JASIC-2024-5-2-06

Fig. 3 Embed File Screen

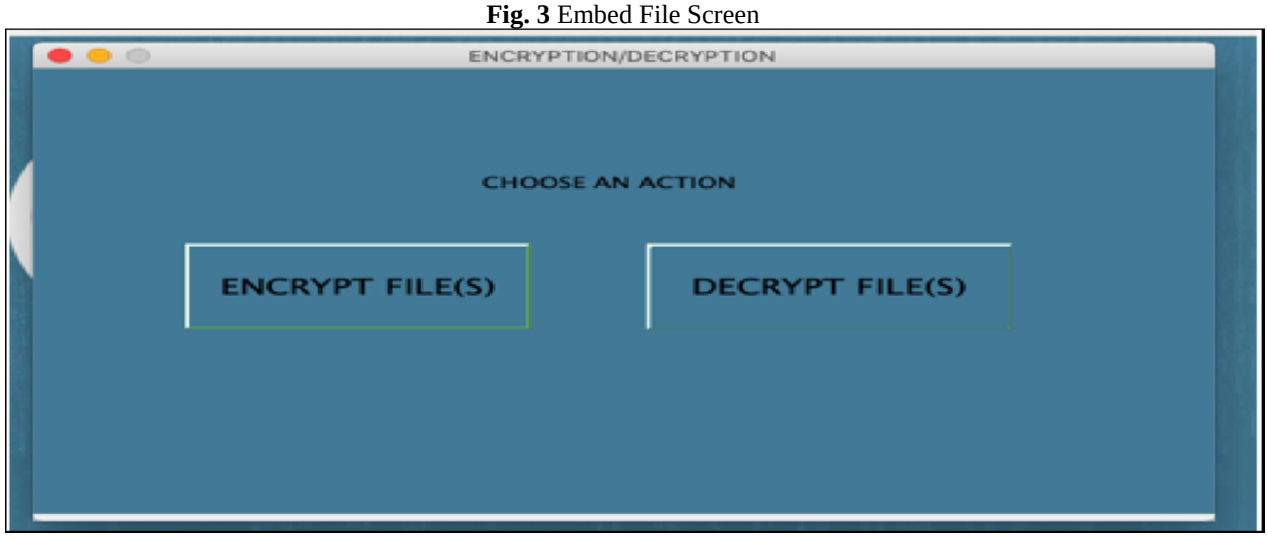

Fig. 4 Encryption and Decryption Menu

Fig. 5 Key Generation Menu

https://doi.org/10.59568/JASIC-2024-5-2-06

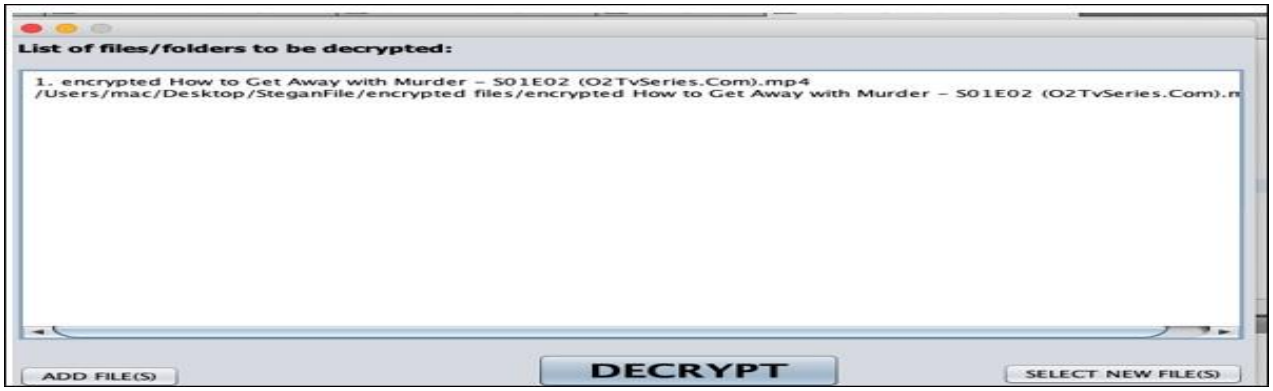

Fig. 6 Decrypt Stego File

Fig. 7 Input Decryption key

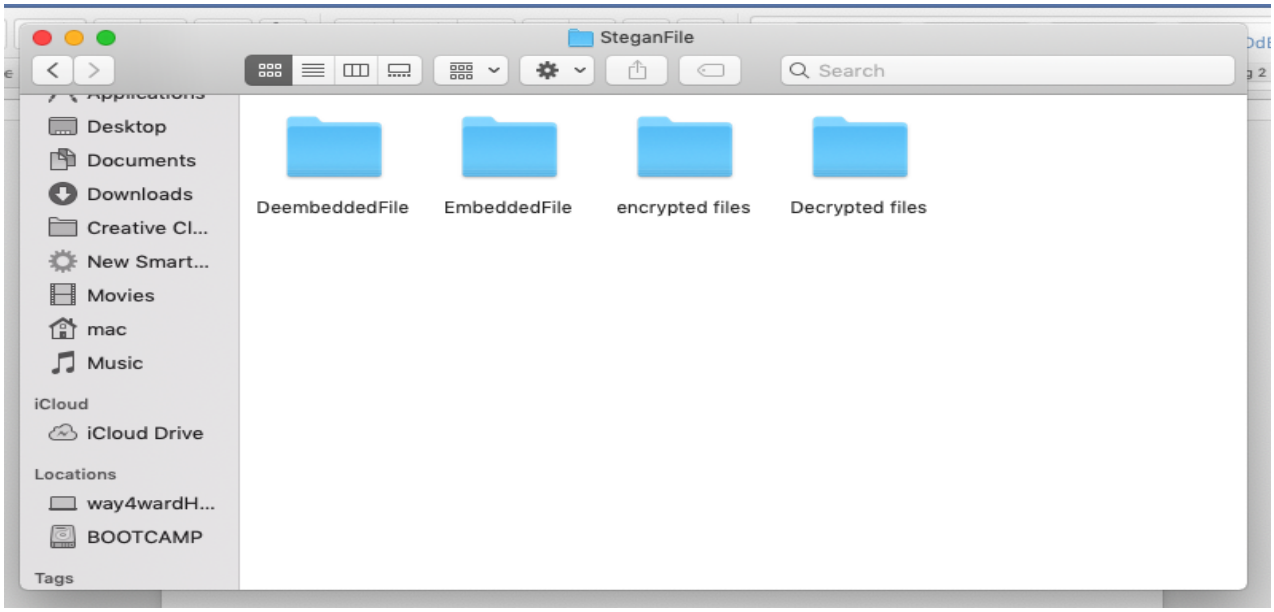

Fig. 8 File Directory

https://doi.org/10.59568/JASIC-2024-5-2-06

4. Results

This section explains the evaluation results obtained when tested using different video sizes; properties and performance distribution of the video files used, wherein it shows the percentage contribution to a whole overtime system execution. The prototype of the system was tested by mimicking peer-assisted networks using different users to represent the peer and CDN server while obtaining the video size, stego-video size, and encryption video size and, encryption and decryption time. Table 1 shows the performance distribution of the video files used.

The ESecureVideo system was tested using selected videos while observing the behavior of the stego video before and after the encryption process. The system shows that the size of the stego video has no typical or definite movement (size), after embedding and encryption, decryption, and de-embedding, the stego file may increase, decrease or remain the same as seen in Table 1. The variation pattern shows a typical behavior of video images during pre-recorded or live Video streaming; this confirms that the ESecureVideo system is not limited to voice data transmission but can effectively secure video contents in the network. Figure 9 depicts the performance analysis of the experiment; it shows a comparison between the original video size, stego-video size, and encrypted-stego-video size. An increase in the size of the video was observed at the stenography phase while the encryption phase produced a relatively low video size. The increased content observed at the stego phase confirms the existence of the secret file where the original video content is hidden. This feature exhibits in the stego phase confirms that the ESecureVideo is capable of securing video content without replacing the original contents after the decryption and de-embedding process. Furthermore, the reduction in video content achieved after encrypting the stego file shows the encrypted-stego-file can be transmitted on the Internet without large bandwidth consumption. This behavior revealed that the ESecureVideo is suitable for large-scale and dynamic network environments such as PA-CDNs.

Table 1 Video Performance

Video size(MB)	Encryption time (ms)	Video size after encryption(MB)	Decryption time (ms)
6.11	156000	6.11	129600
5.03	156000	11.1	129600
4.5	186000	4.5	156000
7.6	186000	12.9	156000
7.1	192000	7.1	168000
5.3	192000	10.9	168000

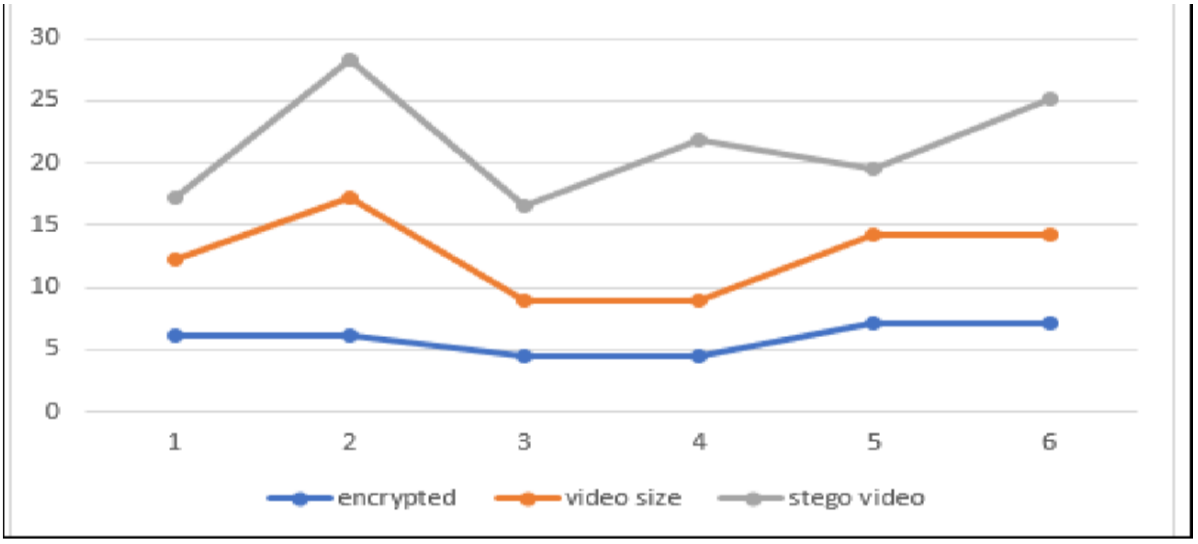

Fig. 9 ESecure Video Evaluation Results

https://doi.org/10.59568/JASIC-2024-5-2-06

5. Conclusion and Future Work

In this research, a secured channel suitable for disseminating multimedia contents, particularly video files is developed. The security model (ESecureVideo) is designed using the principles of steganography and cryptography techniques by ensuring it provides secure video transmission, the ability to retain original video content after de-embedding and decryption process, and providing a secure and alternate channel to transmit private key. The scheme is implemented using java programming language and tested within the peer-assisted network environment. Upon development and testing, it was established that the ESecureVideo model is an efficient mechanism for securing video files and preventing detection of embedded video files.

Furthermore, the proposed security method addresses the integrity of remote file transmission and communication; the system by application mitigates security risks in peer-assisted networks and significantly reduces the incidence of piracy in modern media. It is important to note that the ESecureVideo system is limited to single key generation based on the symmetric cryptography applied. Future work can be focused on using asymmetric cryptography such that private and public keys are obtained for encryption and decryption respectively to improve the system performance. In addition, compressing technique could be incorporated at the steganography phase of the ESecureVideo to further save bandwidth consumption. Lastly, the performance of the system can also be tested with live video streaming applications.

6. References

[1] Abbas, M. S., Mahdi, S. S., & Hussien, S. A. (2020). Security improvement of cloud data using hybrid cryptography and steganography. Paper presented at the 2020 international conference on computer science and software engineering (CSASE).

[2] Abd Zaid, M. M., Ali Talib Al-Khazaali, A., & Abed Mohammed, A. (2024). Lsb steganography using dual layer for text crypto-stego. Paper presented at the BIO Web of Conferences.

[3] Abikoye, O. C., Ojo, U. A., Awotunde, J. B., & Ogundokun, R. O. (2020). A safe and secured iris template using steganography and cryptography. Multimedia Tools and Applications, 79(31), 23483-23506.

[4] Ashari, I. F., Nugroho, E. D., Andrianto, D. D., Yusuf, M. A. N. M., & Alkarkhi, M. (2024). The evaluation of lsb steganography on image file using 3des and md5 key. JITCE (Journal of Information Technology and Computer Engineering), 8(1), 8-18.

[5] Cheng, S. S., Chang, S. L., & Chen, C. Y. (2019, February). Problematic use of live video streaming services: impact of personality traits, psychological factors, and motivations. In Proceedings of the 2019 8th International Conference on Software and Computer Applications (pp. 487-490).

[6] Duan, X., Guo, D., Liu, N., Li, B., Gou, M., & Qin, C. (2020). A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network. IEEE Access, 8, 25777-25788.

[7] Jack, M. M., Gattozzi, D. A., Camarata, P. J., & Shah, K. J. (2021). Live-streaming surgery for medical student education-educational solutions in neurosurgery during the COVID-19 pandemic. Journal of surgical education, 78(1), 99-103.

]8] Khare, P., Singh, J., & Tiwari, M. (2011). Digital image steganography. Journal of Engineering Research and Studies, 2(3), 101-104

[9] Lee, R., Venieris, S. I., & Lane, N. D. (2020, December). Neural Enhancement in Content Delivery Systems: The State-of-the-Art and Future Directions. In Proceedings of the 1st Workshop on Distributed Machine Learning (pp. 34-41).

[10] Majeed, M. F., Ahmed, S. H., Muhammad, S., Song, H., & Rawat, D. B. (2017). Multimedia streaming in information-centric networking: A survey and future perspectives. Computer Networks, 125, 103-121.

[11] Nacakl, S., & Tekalp, M. (2020). Controlling P2P-CDN Live Streaming Services at SDN-enabled Multi-Access Edge Datacenters. IEEE Transactions on Multimedia.

[12] Oduguwa, T., & Arabo, A. (2024). Passwordless authentication using a combination of cryptography, steganography, and biometrics. Journal of Cybersecurity and Privacy, 4(2), 278-297.

[13] Patel, U., Tanwar, S., & Nair, A. (2020). Performance Analysis of Video On-demand and Live Video Streaming using Cloud-based Services. Scalable Computing: Practice and Experience, 21(3), 479-496.

[14]Raj, U. S., & Maheswaran, C. P. (2023). Secure file sharing system using image steganography and cryptography techniques. Paper presented at the 2023 International Conference on Inventive Computation Technologies (ICICT).

[15] Rajabi-Ghaleh, S., Olyaeefar, B., Kheradmand, R., & Ahmadi-Kandjani, S. (2024). Image security using steganography and cryptography with sweeping computational ghost imaging. Frontiers in Physics, 12, 1336485.

 https://doi.org/10.59568/JASIC-2024-5-2-06

[16] Rajamani, K., Srideviponmalar, P., Bebe, P. C., & Samyuktha, C. T. (2021). Secured implementation of steganography in multicloud. Materials Today: Proceedings.

[17]Rodrigues, C. K. D. S., & Rocha, V. (2021). Enhancing BitTorrent for efficient interactive video-on-demand streaming over MANETs. Journal of Network and Computer Applications, 174, 102906.

[18] Sadek, M. M., Khalifa, A. S., & Mostafa, M. G. (2015). Video steganography: a comprehensive review. Multimedia tools and applications, 74(17), 7063-7094.

[19] Salama, G. M., El-Gazar, S., Nassar, R. M., El-Shafai, W., Khalaf, A. A., El-Banby, G. M., . . .El-Samie, F. E. A. (2023). Efficient multimodal cancelable biometric system based on steganography and cryptography. Iran Journal of Computer Science, 6(2), 109-121.

[20] Santiago Lozada, R. E. (2019). Capture the Flag (CTF): Website Tutorial to Boost Cybersecurity Training. Computer Science.

[21] Sarddar, D., Roy, S., & Sen, P. (2017). Edge Multilevel Edge Server Co-operation in Content Delivery Network using Hierarchical Classification. International Journal of Grid and Distributed Computing, 10(3), 41-52.

[22] Siano, P., De Marco, G., Rolán, A., & Loia, V. (2019). A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets. IEEE Systems Journal, 13(3), 3454-3466.

[23] Singh, N. (2017). Survey paper on steganography. International Refereed Journal of Engineering and Science (IRJES), 6(1), 68-71.

[24] Verma, A. (2021). Encryption and decryption of images based on steganography and cryptography algorithms: A new model. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(11), 2839-2846.

[25] Yan, C., Nguyen, Q. N., Benkacem, I., Okabe, D., Nakao, A., Tsuda, T., & Sato, T. (2019, June). Design and implementation of integrated ICN and CDN as a video streaming service. In International Conference on Wired/Wireless Internet Communication (pp. 194-206). Springer, Cham.

[26] Zhang, X, Chen, K., Ding, J., Yang, Y., Zhang, W., & Yu, N (2024). Provably secure public-key steganography based on elliptic curve cryptography. IEEE Transactions on Information Forensics and Security.

 https://doi.org/10.59568/JASIC-2024-5-2-06