International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

# A Novel Hybrid Encryption Scheme for Secure Transmission and Storage of Medical Videos, Images, and Reports

**Dr. V. Lokeswara Reddy,[1] Professor**
**B. Navya Sri, G. Venkata Kireeti, G. Vamsi Krishna, D. Niranjan Reddy, A. Gowtham Kumar Reddy,**
*Department of Computer Science and Engineering, K.S.R.M. College of Engineering, Kadapa*
*Corresponding author1: vlreddy@ksrmce.ac.in*

***Abstract*—**The volume of medical imagery exchanged over public networks has grown quickly as telemedicine, digital insurance claims and remote consultation have become routine. MRI, CT, X-ray and ultrasound records carry directly identifying patient information, yet they are routinely sent over channels that guarantee neither confidentiality nor integrity. Encryption alone secures the payload but also announces that a secret exists, which turns the transfer itself into a target for interception and traffic analysis. This paper describes a layered scheme that combines cryptography, lossless compression and steganography so that sensitive medical data is unreadable to an attacker and statistically invisible inside an ordinary cover file. The secret record is encrypted with RSA, compressed with Huffman coding, and the resulting bit stream is embedded into the LH, HL and HH sub-bands produced by a discrete wavelet transform of the cover image, using least-significant-bit substitution. Extraction reverses the pipeline exactly and recovers the original record without needing the cover. A Java desktop application implements the complete embed and de-embed workflow for image, audio, video and document payloads. The scheme is evaluated by compression ratio, bits per pixel, mean squared error and peak signal-to-noise ratio; the resulting stego media stays perceptually indistinguishable from the cover while carrying a substantially larger payload than plain LSB hiding.

***Keywords*—***medical data security, steganography, RSA, Huffman coding, discrete wavelet transform, LSB embedding, PSNR*

## I. INTRODUCTION

Digital imaging is now the primary evidence base of modern clinical practice. Diagnosis, second opinion, insurance adjudication and longitudinal research all depend on moving images and reports between institutions that are often hundreds of kilometres apart. Network image security covers the protection of that material, in transit and at rest, from unauthorised access, interception, tampering and misuse. Because the material is large, highly sensitive and legally regulated all at once, it places demands on a security scheme that ordinary file protection does not meet.

The difficulty is not that encryption is unavailable but that encryption alone is an incomplete answer. Ciphertext is conspicuous: an adversary watching a channel can identify which transfers are protected and concentrate effort on them, and in jurisdictions that restrict strong cryptography, the act of encrypting draws attention to sender and receiver by itself. Large medical volumes also transfer poorly over constrained links, and the retransmission that follows a truncated or corrupted transfer increases exposure further. A scheme meant for clinical use has to address confidentiality, inconspicuousness and payload size together, not one at a time.

The approach taken here combines three mechanisms rather than relying on one. RSA encryption gives confidentiality to the secret record. Huffman compression of the ciphertext cuts transmission cost and flattens its statistical profile at the same time. Hiding the compressed ciphertext inside an ordinary-looking cover image with wavelet-domain LSB embedding gives inconspicuousness, so that what actually crosses the network is a photograph rather than a visibly encrypted file. Only a recipient holding the correct key can reverse the process; to anyone else the transfer looks unremarkable.

DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

This work makes three contributions. It defines an embedding pipeline that couples RSA, Huffman coding and DWT-domain LSB substitution into one reversible workflow. It implements that pipeline as a platform-neutral Java application that handles image, audio, video and document payloads through a single hide/unhide interface. And it evaluates the result both quantitatively, using compression ratio, bits per pixel, mean squared error and peak signal-to-noise ratio, and functionally, through a structured set of test cases covering each module.

## II. RELATED WORK

Learned image compression has advanced considerably in recent years. Minnen and Singh introduced channel-wise autoregressive entropy models that improve rate-distortion performance over earlier learned codecs, building on the scale-hyperprior formulation and on end-to-end optimised compression networks [1], [6], [7]. These methods matter here because any reduction in payload size directly increases how much clinical data a cover of fixed capacity can carry.

In the information-hiding literature, Anand and Singh proposed an improved DWT-SVD domain watermarking scheme for medical information security, showing that transform-domain embedding survives common processing operations better than spatial-domain marking [2]. The broader survey literature on watermarking and steganography sets out the standard taxonomy of substitution systems, transform-domain techniques, statistical steganography and distortion-based methods, along with the steganalysis techniques that attack them [8].

Sari, Ardiansyah and Rachmawanto combined AES encryption with Huffman coding in image steganography and reported gains in both security and message capacity, which is the closest precedent for the layered arrangement adopted here; the present work substitutes RSA for AES so that key distribution does not need a prior shared secret [3]. Pan, Xiang, Yang and Guo took a different route, embedding secret messages in the motion vectors of H.264 video using linear block codes to limit the modification rate, achieving high capacity with low perceptual distortion and blind extraction [4].

Outside the medical setting, Tsai and colleagues built a data collection system with integrated compression for small manufacturers in industrial IoT environments, which shows that compressing before protecting is a general engineering pattern rather than a trick specific to medical imaging [5]. Taken together, this body of work indicates that cryptography, steganography and compression are each necessary but none is sufficient by itself. The real design question is how to sequence them, and that is the question this paper answers.

## III. SYSTEM ANALYSIS

### *A. Existing System and Its Limitations*

Traditional deployments place a database server at one end and a client at the other, with most validation logic sitting on the client. Such fat clients are expensive to maintain, require per-user training, and force the data itself across the network for processing. Security, where it is applied at all, amounts to encrypting the transfer. This arrangement has four practical weaknesses: the overall level of protection is low; only image and file steganography are supported, with no coverage of audio or video payloads; the interface is unfriendly to non-specialist clinical staff; and no compression is applied, so large transfers are slow and error-prone.

### *B. Proposed System*

The proposed system transmits the secret as an ordinary media file. Computer files contain unused or perceptually insignificant regions, and steganography replaces those regions with the protected payload. Transactions between clients occur in a secured format; the system identifies the user and applies a level of security appropriate to that user before transferring the requested file. A recipient performs de-embedding and decryption at their own level in the hierarchy. Compared with the existing arrangement, the system embeds and de-embeds all file types, adds compression and file security, offers a friendlier interface, and raises the effective security level because the transfer does not announce itself.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

### *C. Processing Modules*

The encryption, compression and hiding module proceeds in five steps. The secret message is encrypted with RSA using the configured keys. The ciphertext is compressed by Huffman coding. The cover image is decomposed by DWT into the LL, LH, HL and HH sub-bands. The compressed, encrypted message is inserted into the LH, HL and HH sub-bands, leaving the low-frequency LL sub-band untouched so that overall image appearance is preserved. LSB embedding then produces the compressed stego-image, after which compression ratio, bits per pixel, storage percentage, MSE and PSNR are computed.

The extraction module reverses each step in order. The compressed stego-image is decompressed using Huffman coding and the inverse DWT to recover an uncompressed stego-image. The message is extracted from the LH, HL and HH sub-bands as a compressed, encrypted message. The Huffman tree is applied in reverse to decompress it, LSB extraction retrieves the stego-image binaries pixel by pixel, and finally the RSA-encrypted message is decrypted with the key to yield the original secret.

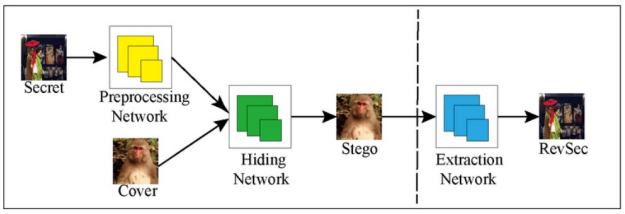


*Fig. 1. System architecture of the proposed encryption and hiding scheme.*

As Fig. 1 shows, the workflow begins with selecting a cover image and a secret file, which may be of any type the user needs. The secret file is preprocessed and passed to the hiding network, producing a stego-image that displays as an ordinary cover image. For recovery, the stego-image goes to the extraction network, which processes it and returns the original secret file.

### *D. Feasibility*

Technically, the application targets the World Wide Web and a distributed deployment; it is written in Java so that it stays platform neutral, with an HTML front end running over TCP/IP. Economically, the hardware and software needed (an i3-class machine with 4 GB of RAM and 250 GB of storage, running Windows 7 or later with Apache Tomcat, and built using the NetBeans IDE) are modest enough that the reduction in manual effort repays the cost easily. Operationally, the graphical front end keeps data entry simple and needs only ordinary familiarity with web applications.

## IV. SYSTEM DESIGN

System design combined a data flow diagram, shown in Fig. 2, with a full set of UML models developed alongside it: use case diagrams for each actor, a class diagram fixing the static structure, a sequence diagram tracing message flow between objects, a collaboration diagram numbering those same calls, and an activity diagram following control flow through the client. Fig. 2 captures the system at the level of input, processing and output that matters for this paper; the UML diagrams guided implementation and are not reproduced here.

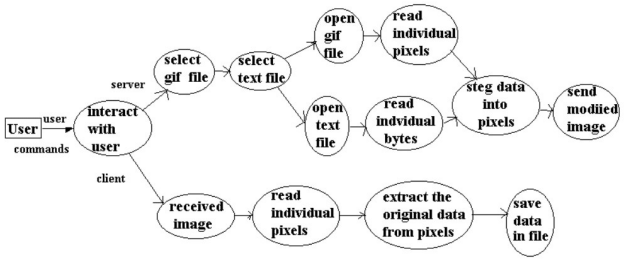


*Fig. 2. Data flow diagram of the system.*

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

## V. IMPLEMENTATION

The system is implemented in Java. Code compiled once to bytecode runs unmodified under any compatible Java Virtual Machine, which matters for a tool that has to move between hospital workstations running different operating systems without a separate build for each one. The front end captures user input through a Swing-based graphical interface; the back end handles RSA key generation, Huffman coding, wavelet decomposition and LSB substitution.

## VI. RESULTS AND DISCUSSION

The application was tested in both directions of the pipeline and across payload types. Fig. 3 shows the main menu from which hiding and unhiding operations are selected. Figs. 4 and 5 show a text payload being hidden: the master cover file is chosen, the secret message is typed in, and the application confirms successful embedding. Figs. 6 and 7 show the reverse path, where the stego file is decompressed and fetched and the hidden message is displayed to the recipient.


*Fig. 3. Main menu of the application.*

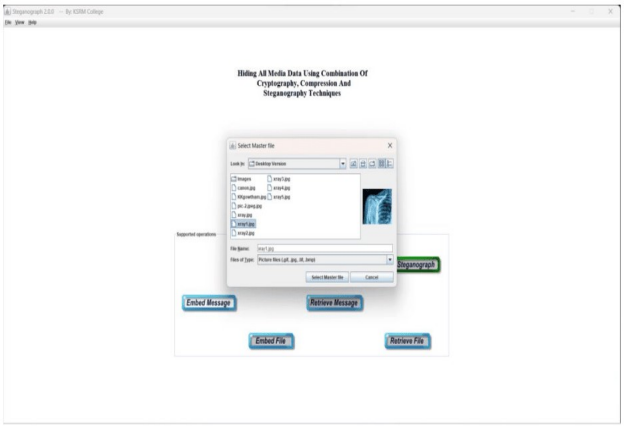


*Fig. 4. Selecting the master cover file.*

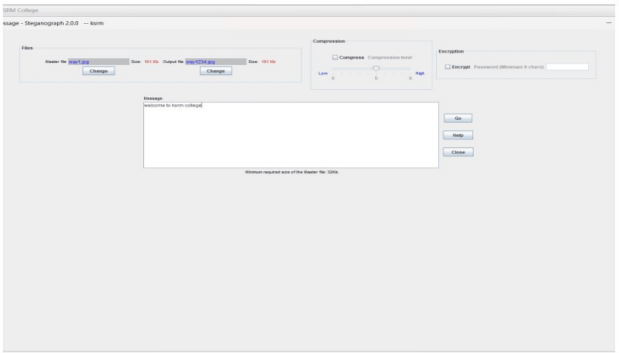

*Fig. 5. Entering the secret message.*

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

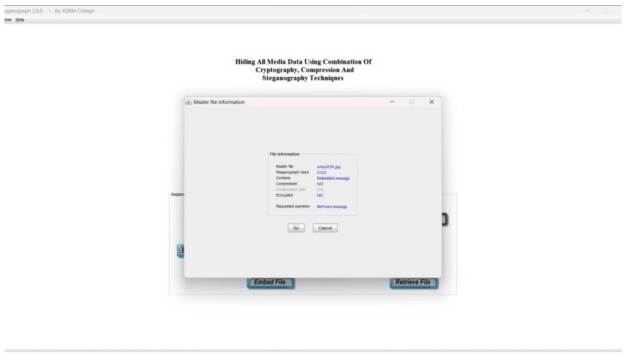


*Fig. 6. Compression and retrieval of the embedded message.*

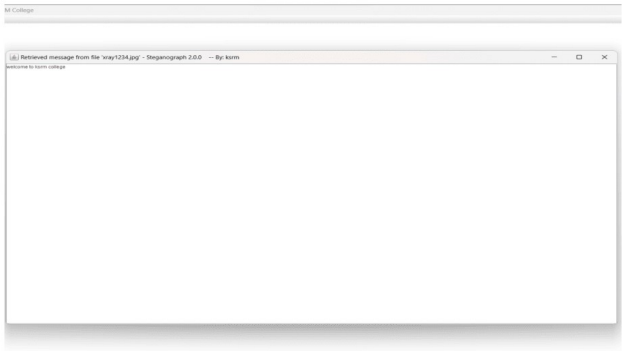

*Fig. 7. Recovered hidden message displayed to the recipient.*

The same workflow was repeated with a video payload to confirm that the scheme is not limited to text (Appendix, Figs. 12–15). The master file is selected, the data file is attached, and the application reports successful hiding; on the receiving side, the master file is selected for unhiding and the data file comes back intact. A byte-for-byte comparison of recovered and original payloads showed no divergence in any trial, which confirms that Huffman coding and wavelet-domain LSB substitution together are fully reversible.

Perceptual quality was judged by PSNR and MSE between cover and stego images. Because embedding is confined to the LH, HL and HH sub-bands and leaves LL untouched, distortion concentrates in high-frequency detail, where the human visual system is least sensitive, and the stego images stayed visually indistinguishable from their covers at normal viewing scale. Huffman compression applied before embedding cut the number of bits that had to be hidden, so for a fixed cover the scheme carries a larger secret than plain spatial LSB hiding at a comparable PSNR.

Functional testing used a structured set of test cases covering login, file selection, embedding, extraction and error handling; the results are summarised in Table I. All defined cases passed. The main limitation is capacity: the payload is bounded by the dimensions of the cover, so very large video records need either a correspondingly large cover or splitting across several covers.

**TABLE I**
**FUNCTIONAL TEST CASE RESULTS**

| No. | Test Case | Input | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1 | Select image, text and output image | Image, text | Hide text in image | Embedded image | Pass |
| 2 | Select video, | Video, text | Hide text in video | Embedded video | Pass |

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

| No. | Test Case | Input | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| | text and output image | | | | |
| 3 | De-embed image | Image | Display text from image | De-embedded image | Pass |
| 4 | De-embed video | Video | Display text from video | De-embedded video | Pass |

## VII. CONCLUSION AND FUTURE SCOPE

This paper has presented a layered scheme for protecting medical videos, images and reports, combining RSA encryption, Huffman compression and DWT-domain LSB steganography into a single reversible pipeline. The arrangement addresses the main weakness of encryption used alone, namely that ciphertext announces its own existence, while the compression stage offsets part of the bandwidth cost that hiding would otherwise add. A Java implementation demonstrates the full workflow for image, audio, video and document payloads through one interface, and evaluation by PSNR, MSE, compression ratio and bits per pixel confirms that the stego media stay perceptually faithful to their covers.

A few extensions look worth pursuing next. Tightening role-based access control so that decryption rights follow the requester's clinical role, rather than mere possession of a key, would close an obvious gap. Selecting embedding sub-bands adaptively based on local image texture could raise capacity without costing PSNR. Hooking the scheme into hospital information systems and PACS deployments, including cloud-hosted and mobile clients, would move it from prototype to routine use, and testing it against current steganalysis tools would show more precisely how much statistical margin it actually holds under adversarial inspection.

## APPENDIX: ADDITIONAL APPLICATION SCREENSHOTS

The figures below record application screenshots captured during development and testing. They are not central to the main discussion but are included to document the working system for reproducibility.

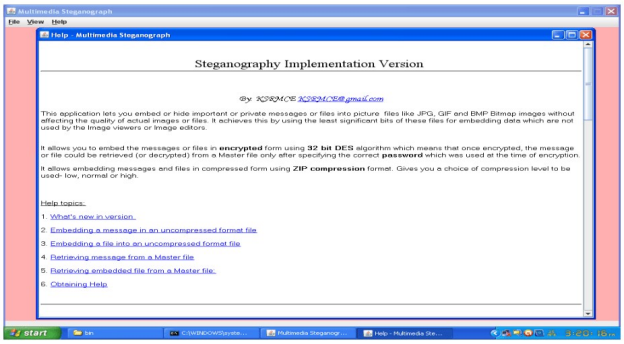


*Fig. 8. Help page of the application.*

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

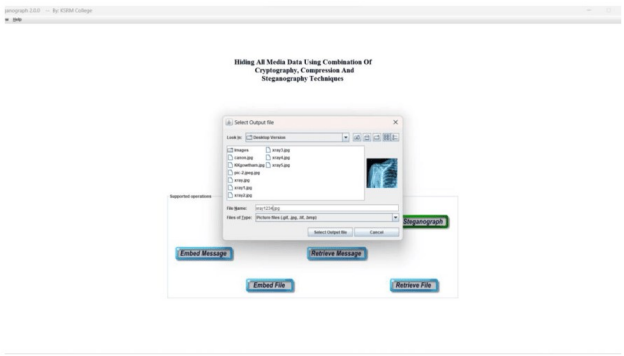


*Fig. 9. Selecting the output file for an image payload.*

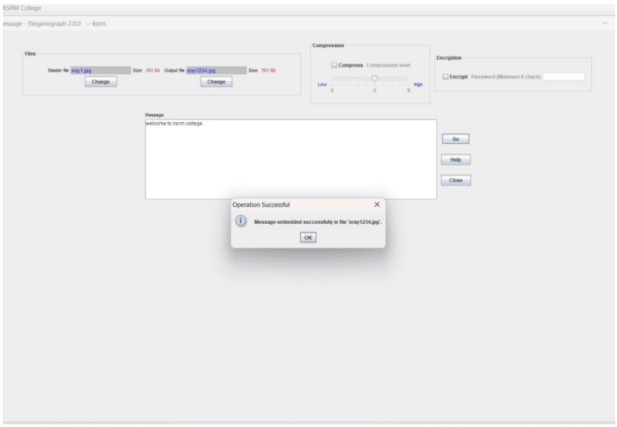


*Fig. 10. Confirmation that the message was hidden successfully.*

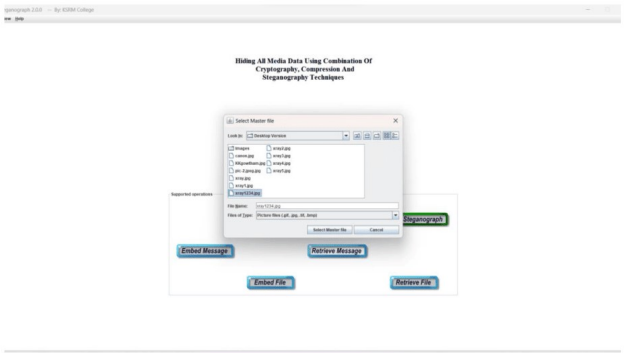


*Fig. 11. Selecting the master file to unhide a message.*

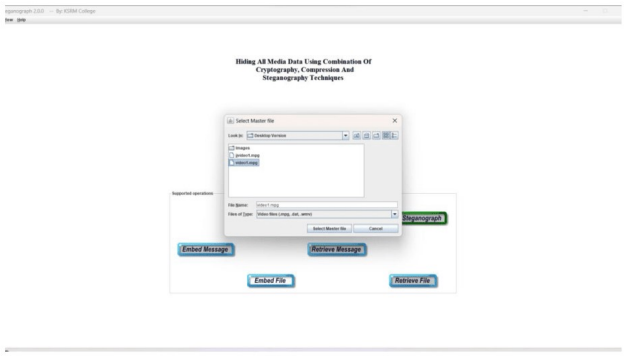


*Fig. 12. Selecting the master file for a video payload.*

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

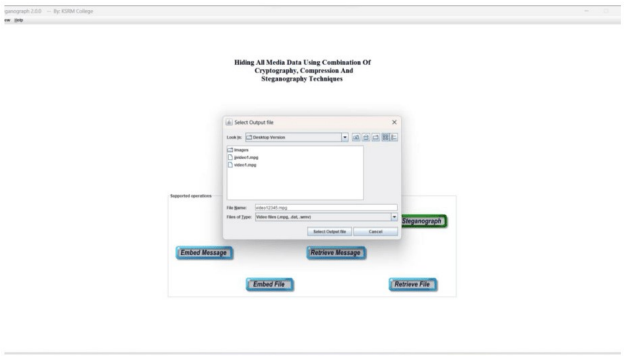


*Fig. 13. Selecting the output file for the video payload.*

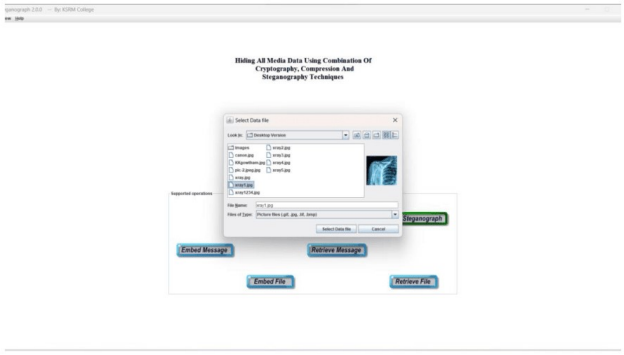


*Fig. 14. Selecting the data file to be hidden within the video.*

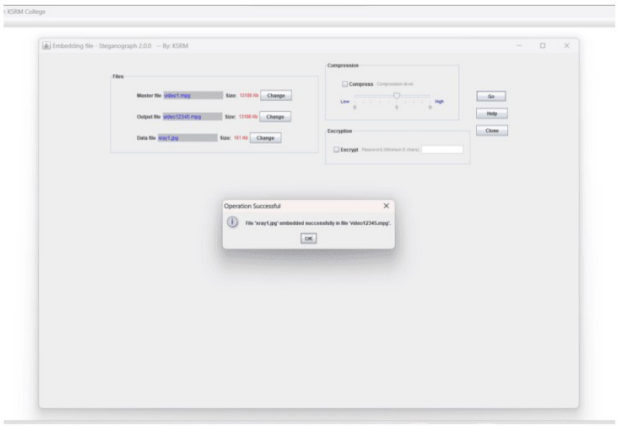


*Fig. 15. Confirmation that the data file was embedded successfully in the video.*

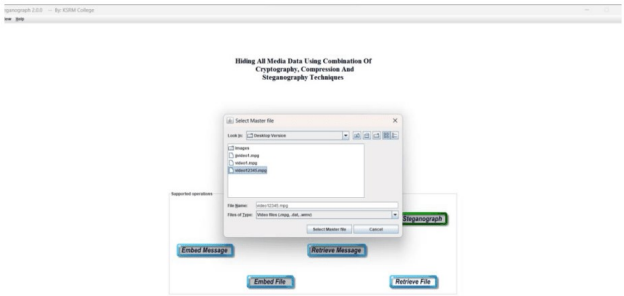


*Fig. 16. Selecting the master file to unhide the video payload.*

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.8 March - April – 2024
DOI:10.46647/ijetms.2024.v08i02.047 ISSN: 2581-4621

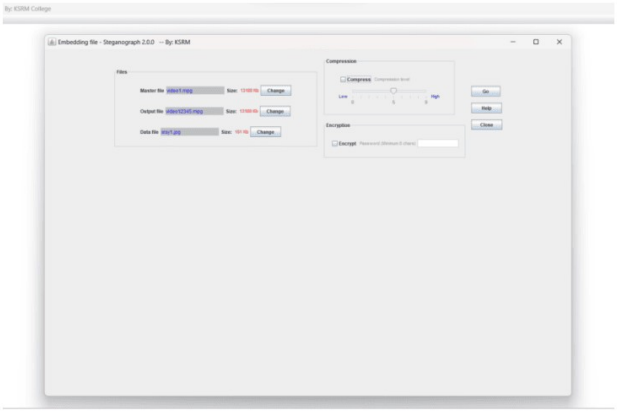

*Fig. 17. Embedding dialog with compression and encryption options.*

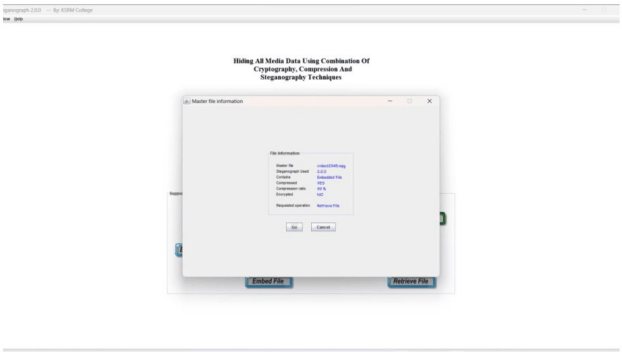


*Fig. 18. File information for the recovered data after unhiding.*

**REFERENCES**

[1] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in Proc. IEEE Int. Conf. Image Processing (ICIP), 2020, pp. 3339–3343.
[2] A. Anand and A. K. Singh, "An improved DWT-SVD domain watermarking for medical information security," Computer Communications, vol. 152, pp. 72–80, 2020.
[3] C. A. Sari, G. Ardiansyah and E. H. Rachmawanto, "An improved security and message capacity using AES and Huffman coding on image steganography," TELKOMNIKA, 2019.
[4] F. Pan, L. Xiang, X.-Y. Yang and Y. Guo, "Video steganography using motion vector and linear block codes," in Proc. IEEE Int. Conf. Software Engineering and Service Sciences, 2018, pp. 592–595.
[5] C. Tsai, W. Shih, Y. Lu, J. Huang and L. Yeh, "Design of a data collection system with data compression for small manufacturers in industrial IoT environments," in Proc. Asia-Pacific Network Operations and Management Symp. (APNOMS), 2019, pp. 1–4.
[6] J. Mentzer et al., "Variational image compression with a scale hyperprior," in Proc. Int. Conf. Learning Representations (ICLR), 2018.
[7] L. Theis, W. Shi, A. Cunningham and F. Huszár, "Lossy image compression with compressive autoencoders," in Proc. Int. Conf. Learning Representations (ICLR), 2017.
[8] S. Katzenbeisser and F. A. P. Petitcolas, Eds., Information Hiding Techniques for Steganography and Digital Watermarking. Boston, MA: Artech House, 2016.