Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.068429

ARTICLE

Lightweight Multi-Layered Encryption and Steganography Model for Protecting Secret Messages in MPEG Video Frames

Sara H. Elsayed[1], **Rodaina Abdelsalam**[1], **Mahmoud A. Ismail Shoman**[2], **Raed Alotaibi**[3,*] and **Omar Reyad**[4,5,*]

[1]Faculty of Computers and Information Technology, EELU, Giza, P.O. Box 12611, Egypt
[2]Faculty of Computers and Artificial Intelligence, Cairo University, Giza, P.O. Box 12613, Egypt
[3]Applied College, Shaqra University, Shaqra, 11961, Saudi Arabia
[4]College of Computing and Information Technology, Shaqra University, Shaqra, 11961, Saudi Arabia
[5]Faculty of Computers and Artificial Intelligence, Sohag University, Sohag, P.O. Box 82524, Egypt

*Corresponding Authors: Raed Alotaibi. Email: alhafi@su.edu.sa; Omar Reyad. Email: oreyad@su.edu.sa

Received: 28 May 2025; Accepted: 30 July 2025; Published: 23 October 2025

ABSTRACT: Ensuring the secure transmission of secret messages, particularly through video—one of the most widely used media formats—is a critical challenge in the field of information security. Relying on a single-layered security approach is often insufficient for safeguarding sensitive data. This study proposes a triple-lightweight cryptographic and steganographic model that integrates the Hill Cipher Technique (HCT), Rotation Left Digits (RLD), and Discrete Wavelet Transform (DWT) to embed secret messages within video frames securely. The approach begins with encrypting the secret text using a private key matrix (PK_1) of size 2 × 2 up to 6 × 6 via HCT. A second encryption layer is applied using a dynamic private key (PK_2) derived from the RGB pixel values of the video frame, resulting in a rotated cipher. The doubly encrypted message is then embedded into the video frames using the DWT method. Upon transmission, the concealed message is extracted using inverse DWT and decrypted in two steps—first with PK_2 and then with the inverse of PK_1. Experiments conducted using MPEG video sequences and message lengths ranging from 10 to 300 bytes demonstrate strong performance in terms of Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Correlation Coefficient (CC) between original and encrypted messages. The similarity between original and stego frames is further validated using Structural Similarity Index (SSIM), Mean Absolute Error (MAE), Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI). Results confirm that utilizing video frames to generate PK_2 offers superior security compared to static key images. Moreover, the indistinguishability between original and stego frames highlights the method's robustness against visual and statistical attacks.

KEYWORDS: Lightweight encryption; stego frame; HCT; RLD; DWT; SSIM; MAE; NPCR; UACI

1 Introduction

Multimedia is a media that allows information to be easily transferred from one location to another. Multimedia security has become more important because it plays a key role in securing the content of multimedia over the internet [1]. It has therefore been recommended that multimedia content be secured for individuals, private enterprises, and the government [2,3]. The proliferation of digital video content in various sectors, including entertainment, education, and communications, has made security a critical concern [4]. With the increasing availability of high-speed internet and the widespread use of digital devices, digital video distribution has grown exponentially. However, this expansion has also exposed video content

to various security threats, such as unauthorized access, piracy, and tampering [5]. A content protection technique is defined as a procedure to protect multimedia content from illegal user access. Content protection mechanisms, such as encryption, Digital Rights Management (DRM), and watermarking, have traditionally been employed to secure both digital images and videos [6]. While these methods are effective in many cases, they often involve substantial computational overhead and complexity, which can be resource-intensive. Additionally, many of the vulnerabilities found are difficult to resolve and require considerable effort to investigate; as a result, they often need to be addressed for extended periods, leaving Android users vulnerable to malicious attacks [7]. Given the evolving nature of threats and the diversity of user devices, there is a growing need for lightweight secret message protection models that strike a balance between security, performance, and resource efficiency. Chaos theory has enabled the development of advanced encryption techniques across various industries, enhancing data security [8]. Numerous computational methods—such as heuristics, clustering, deep learning, evolutionary algorithms, and frequent pattern mining—have been employed to support these advancements [9]. A lightweight model should minimize computational requirements while providing robust data hiding against common threats, including unauthorized duplication, tampering, and redistribution. In the realm of digital media, video file formats play a critical role in determining the quality, compression, and compatibility of video files across different devices and platforms. Two widely recognized video formats are Moving Picture Experts Group (MPEG) and Audio Video Interleave (AVI) [10]. Both formats have been essential in the evolution of digital video, each offering unique advantages and applications. When comparing the security features of MPEG and AVI video formats, MPEG offers far superior protection mechanisms [11]. MPEG's support for encryption, DRM, watermarking, and secure streaming makes it the preferred choice for protecting high-value digital content from piracy and unauthorized access. AVI, while flexible and widely compatible, lacks these built-in security features, making it less suitable for secure content distribution, particularly over the internet.

The video is a collection of frames displayed in rapid succession, creating the illusion of motion. Each frame is considered a standalone image that can be processed, analyzed, and manipulated independently. The proposed method changes the video size and extracts RGB values from certain video frames, then resizes the resulting matrix from one of these color values (R, G, or B). Each color value ranges from 0 to 255, represented in one row with multiple columns equal to the secret message size. This resized matrix is then used as a private key (PK) in the cryptographic process. The first index value in the generated PK is used in the RLD calculation for all the divided blocks. There are several important points to consider when applying this method:

- Changing the frame in encryption or decryption will change the used pixel values and so the decrypted message.
- Changing the PK values will affect the secret message, which may result in the wrong one.
- Larger video sizes lead to a higher number of video frames.
- Larger frame dimensions increase the time required to extract the frames.

Considering these points, the proposed method uses smaller video sizes and lower frame dimensions, as larger frame dimensions would require significantly more time to extract all video frames. So, the presented method uses HLT, RLD, and DWT techniques to obtain a robust model to protect secret messages inside video frames. The used video data file is available through the reference paper [12].

The main contributions of this work can be summarized as follows:

- Proposed a lightweight security hybrid model for secret message protection utilizing MPEG digital video frames.
- Key generation, encryption, and decryption times are computed for the proposed security method.

- Comparison between AVI and MPEG based on RLD encryption calculations is provided.
- Comparison of encryption and decryption times of the proposed method based on different video sizes and frame dimensions.
- MSE, PSNR, and CC (cipher and plaintext comparison) for different video sizes and frame dimensions.
- SSIM, MAE, NPCR, and UACI comparison between original and stego frames is also provided.
- Comparison of the proposed work according to MSE, PSNR, and CC between the original and encrypted messages is also reported.

The rest of the paper is structured as follows. Section 2 presented preliminaries about the triple lightweight model. The related work on video security techniques is covered in Section 3. In Section 4, the proposed method is presented. The experimental results are provided in Section 5. The paper's conclusions and future actions are outlined in Sections 6 and 7.

2 Preliminaries

2.1 Hill Cipher Technique

The Hill cipher belongs to the class of polygraphic substitution ciphers and is a symmetric key algorithm [13]. It divides the secret text into groups of an equal number of blocks, depending on the key matrix m ∗ m. By giving each letter a numerical value, the normally readable text is divided into tom-blocks that satisfy the key matrix K size [14]. Key matrix is designed for some conditions, such as:

Determiner not equal to zero and its value is a number that must be coprime to 256 (as dealing with ASCII characters).

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{n1} & k_{n2} & k_{n3} & \dots & k_{nn} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{pmatrix} \bmod 256, \tag{1}$$

where p is the plaintext represented as (p_1, p_2, p_3, …, p_n) and k is the key matrix (k_1, k_2, k_3, …, k_n). Multiplication Result will obtain the ciphertext in the encryption phase. The decryption phase requires the key inverse matrix K^{-1}, which means that $K * K^{-1} = I$, where I is the identity matrix.

$$C = K * P \bmod 256 = (C1, C2, C3, \dots, Cn) \tag{2}$$

The decryption phase requires the key inverse matrix K^{-1}, which means that $K * K^{-1} = I$, where I is the identity matrix.

$$P = K^{-1} * C \bmod 256 \tag{3}$$

2.2 Rotation Left Digits

The rotation left digits (RLD) is a mathematical and computational operation that involves cyclically shifting the digits of a number [15]. In a left rotation, the leftmost digits are moved to the rightmost position, while the remaining digit is shifted to the left as shown in Fig. 1.

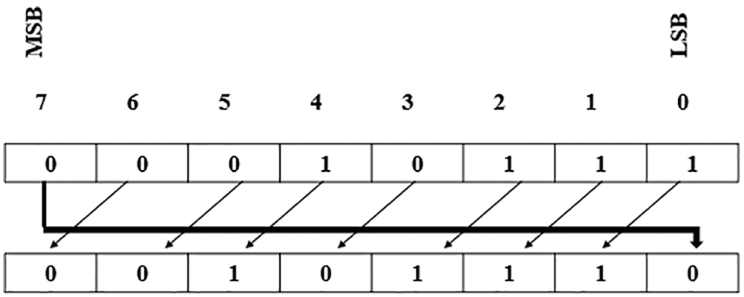

Figure 1: Rotation left digits operation

In cryptography, rotating digits are used as part of various encryption algorithms to obfuscate data, making it harder for unauthorized parties to interpret [16]. Left digit rotations can add complexity to encrypted data without introducing significant computational overhead. The technique of Least Significant Bit (LSB), which is applied to rotate its value, refers to the lowest order bit. The main advantage of LSB compared to other techniques is that it is easier to hide secret message bits directly in the least significant bit plane of the cover image [17]. The user can recover the secret message by extracting the least significant bits of the manipulated pixels to recover the original message. The Most Significant Bit (MSB) method refers to the highest pixel value in a binary number. Strong cryptographic systems ensure that changes in the MSB (or any bit) produce unpredictable outcomes in encryption, ensuring security against brute-force or bit-level attacks [18]. It has a critical role in ensuring the integrity and security of cryptographic operations, particularly where binary manipulation is involved.

2.3 Discrete Wavelet Transform

Steganography techniques involve spatial domain techniques, in which the strength of the pixel conceals the data. The value of the pixel changes as the data is embedded. Data is concealed in a frequency with a broad bandwidth using the spread spectrum technique [19]. However, as hiding strategies evolved, so did methods to try to remove their hiding and hack their contents; as a result, encryption and hiding techniques were merged to boost the effectiveness of these approaches. Steganography is a set of techniques for concealing information using multimedia data, such as image, text, audio, video, and network [20]. It is crucial to conceal information from attackers in video files by using intermediary channels like AVI and MPEG without compromising the content of the containers. To ensure that the secret media is not apparent after the confidential data is inserted into the cover video file, the authenticity of the hidden files shouldn't be shared or disseminated [21]. A mathematical tool for hierarchically breaking down a video frame is the wavelet domain technique. It is predicated on tiny waves, known as wavelets, with variable frequencies and brief durations. The 2-D frame is processed using 2-D filters in both dimensions when DWT is applied. It operates both horizontally and vertically. The vertical filter splits the frame in half, storing the average coefficients in the first half and the detail coefficients in the second [22]. Frame pixels are converted to wavelets via DWT. Better quality frames with a higher compression ratio are provided by DWT's lossless frame compression [23]. To make the difference between the original and embedded files undetectable, the secret data must be embedded into an MPEG file. High embedding rates without sacrificing video quality are possible with the DWT method. It has strong characteristics that set it apart from encryption techniques. Information hiding is a crucial technique for securing sensitive and important data, especially when it comes to data transmitted through various digital communication channels. Its foundation is the idea of hiding data inside other digital media as a carrier cover.

3 Related Work

Traditional video security techniques may involve either full or partial encryption. However, most research articles do not address text encryption through pixel values of frames. Additionally, many studies utilize the AVI extension for this purpose, employing steganography techniques to embed cover frames into stego videos. Steganography and watermarks are two well-known security techniques that are typically used to hide information. They both alter the media file's content (image, video, and audio) directly for copyright protection and secret communication or sender identification [24]. The authors in [25] presented a method that tends to be quite complex when restricted to AVI video files, particularly without consideration for the execution time associated with this approach. Moreover, it is unsuitable for specific scenarios, such as when working with compressed files or smaller file sizes. The AVI format has some disadvantages compared to MPEG. MPEG offers a higher frame rate than AVI, and it employs more efficient codecs, resulting in larger file sizes and fewer frames. A neural network model is employed to extract video features for steganography as presented in [26]. It tends to increase time complexity and may not yield the most effective results in extracting hidden messages. This efficiency aids in compressing video data without quality degradation. The Visual Geometry Group (VGG) model achieves high accuracy in [27]. There is insufficient information regarding its time complexity or resistance to attacks. While techniques such as zigzag scanning and the LSB method are used to conceal secret messages, details regarding the specific algorithms and efficiency are not provided in [28]. Researchers in [29] rely solely on 256 × 256 greyscale input images for testing; however, this image type has a limited pixel value range, with 0 representing black and 255 representing white. In contrast, color images present a broader range of pixel values from 0 to 255 for each RGB channel. The work in [30] does not clearly measure its coefficients and does not consider execution time. Although the authors in [31] employ Deoxyribonucleic Acid (DNA) rules, time complexity is not mentioned. The study in [32] generates image keys using non-uniform cellular automata, but the performance regarding encryption speed requires further analysis. In [33], despite employing Discrete Cosine Transform (DCT) for encrypting Multimedia Video Coding/High-Efficiency Video Coding (MVC/HEVC) data, the execution time is relatively high at approximately 0.4 s per one ciphered frame, indicating a need for improvement. The execution time in [34] varies between 4 and 6 s, which is considered very high, necessitating enhancements. The work in [35] employs chaotic maps to produce a keystream; however, the average execution times are 2 and 3 s, which are considered somewhat high, indicating a need for improvement in this area.

Using a contemporary modified blowfish technique, the authors in [36] showed that the encryption time could be increased to 124.1 ms and the decryption time to 124.3 ms. In general, the ideal execution time should be around a thousandth of a second, aligning with the decryption time. A general comparison between AVI and MPEG indicates the suitability of MPEG video type for large frame numbers. For the same video file size, MPEG typically accommodates approximately twice as many frames as AVI, highlighting its superior compression efficiency and suitability for longer video sequences, as illustrated in Table 1.

Table 1: Comparison between AVI and MPEG according to frame rate, codec type, and frame number

Video type	Frame rate	Codecs	Frame number
MPEG	30 or 60 frames per second	H.264 or HEVC	Video frames
AVI	24 or 25 frames per second	Older or fewer codecs	Half of the video frames

4 Proposed Method

The proposed method splits into three parties. First, the secret text message is encrypted using the Hill Cipher key matrix PK_1 according to the message length. Private Key matrix PK_1 starts from $2 * 2$ to $6 * 6$ as shown in Table 2.

Table 2: First private key matrix (PK_1) length used in the proposed method

Message length (Bytes)	Key matrix length
10	$2 * 2$
20	$2 * 2$
30	$3 * 3$
60	$3 * 3$
70	$2 * 2$
80	$4 * 4$
100	$4 * 4$
150	$5 * 5$
180	$4 * 4$
200	$5 * 5$
250	$5 * 5$
300	$6 * 6$

Secret message is encrypted first using Hill cipher key matrix resulting encrypted Hill message, then the encrypted Hill message will be encrypted for the second time using rotation left digits RLD_e (rotations number in encryption phase) and second private key PK_2 which generated from a video frame pixel values, resulting rotated encrypted message, then this rotated encrypted message will be embedded using DWT technique inside the video frame. According to video frames, divide the color video frames into the appropriate color channels (R, G, and B). Each frame's red, green, and blue channels are represented by three distinct 2D matrices. Each color channel is subjected to DWT individually, which breaks each channel down into detail and approximation coefficients at various levels. The secret message will then be embedded in the color video frames by altering the wavelet coefficients of each color according to its binary values. Inverse DWT is applied to each color channel once the coefficients have been changed to reconstruct the color frames.

4.1 Method Framework

The proposed method depends on the Hill Cipher technique and MPEG digital video frames because of the AVI defects, as well as the number of rotations in MPEG is higher than AVI, as shown in Table 3.

Table 3: Comparison between AVI and MPEG (RLD_e) encryption calculations to video frames dimensions

Message size (Bytes)	RLD_e encryption calculations					
	Frames dim [240, 320]		Frames dim [360, 480]		Frames dim [600, 1050]	
	AVI	MPEG	AVI	MPEG	AVI	MPEG
10	79	118	79	117	79	118
20	122	174	122	154	123	175

(Continued)

Table 3 (continued)

Message size (Bytes)	RLD_e encryption calculations					
	Frames dim [240, 320]		Frames dim [360, 480]		Frames dim [600, 1050]	
	AVI	MPEG	AVI	MPEG	AVI	MPEG
30	189	274	189	244	189	214
50	140	237	140	183	140	219
70	116	303	116	231	116	199
80	133	282	133	230	133	229
100	211	322	211	218	211	223
150	269	387	269	333	269	330
180	291	542	291	342	291	245
200	316	598	316	362	316	362
250	290	691	290	396	290	390
300	425	639	465	425	425	521

The secret message is encrypted first by PK_1, which is generated from the Hill key matrix. The MPEG video file is then resized and split into frames. A different frame is chosen for each text message that is between 10 and 300 bytes long to make sure that the RGB pixel values are different. The RGB values for each frame are then recovered. Then select one of these RGB values as the PK_2 that changes dynamically with the content of the video based on the secret message size, either R or G, or B. PK_2 serves as a content-aware cryptographic enhancement that's hard to guess or replay. The generated PK_2 and encrypted Hill secret message length (L) are then used to calculate rotation left digits' operation, where $RLD_e = (L/2) * 4 + PK_2(I) + C$ in the encryption phase which I indicating the first generated private key (PK_2) value and C is a constant refers to the rotation of 1-MSB value as the reminder seven bits refer to the LSB rotations as mentioned in Fig. 1. This process results in a rotated encrypted message, which can be hidden inside the selected video frame using the DWT technique as depicted in Fig. 2.

After the transmission of the stego frames over communication channels, the hidden message is retrieved from the stego frame, and the same PK_2 is used again to decrypt it. The RLD_d calculation in the decryption phase will be a bit different, where $RLD_d = 2 * L * 4 - RLD_e$, which will result in a decrypted rotated message, then this message is decrypted again with the Hill key matrix inverse $(PK_1)^{-1}$, resulting in the original text message.

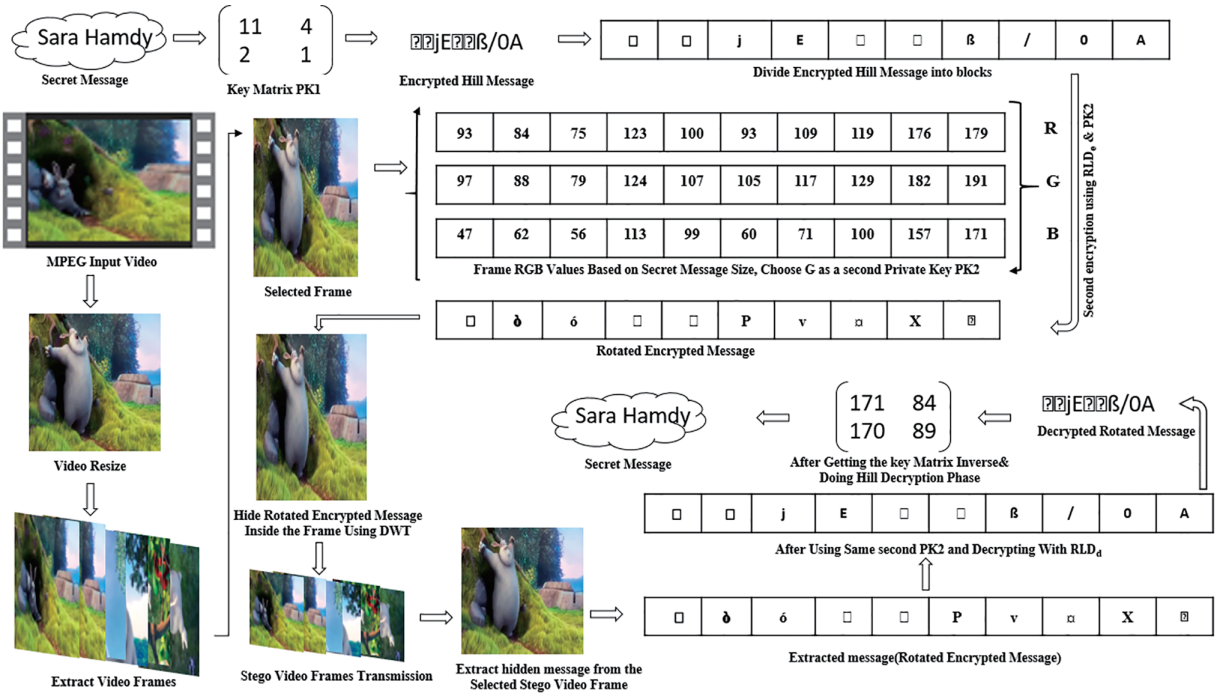

Figure 2: Proposed work architecture

4.2 Encryption and Decryption Algorithms & Flowcharts

A Hill key matrix (PK_1) is obtained along with the secret text message, which is initially encrypted using the Hill cipher algorithm. A frame-by-frame division and resizing of the MPEG video file follow. RGB values are then obtained for each frame. PK_2 is created using RGB pixel values that, depending on the size of the hidden message, will dynamically alter with the content of the video. The rotating encrypted message can be obtained by encrypting each sub-block. After utilizing DWT to conceal the resulting rotated encrypted message within the video frame, the stego frame is saved as represented in Algorithm 1.

Algorithm 1: Encryption and hiding

1. Get a secret text message
2. Get the key matrix (PK_1)
3. Encrypt the secret text message using the Hill cipher technique
4. Get the encrypted Hill message
5. Get MPEG video
6. Resize the MPEG video
7. Extract video frames
8. Generate the (PK_2) from RGB pixel values
9. Divide the encrypted Hill message into a number of sub blocks equal to the (PK_2) length L
10. Compute $RLD_e = (L/2) * 4 + PK(I) + C$
11. For all the sub blocks, do
    a. Transform each sub block to binary
    b. Reshape the binary digits into one row matrix

(Continued)

Algorithm 1 (continued)

c. Rotate the one row matrix RLD_e times to the left
d. Transform the resulting matrix to a string
e. Get the resulting encrypted message
End for
12. Hide the rotated encrypted message inside the video frame using DWT
13. Save the stego frame

The concealed rotated encrypted message must first be retrieved in order to retrieve the original message from the stego video frame. The rotated encrypted message is split into blocks of the length (L) of PK_2, using the same second PK_2 as during the encryption phase. To produce the decrypted rotated matrix, the binary representation of each rotated encrypted block is translated into a one-row matrix, rotated left by RLD_d times, and then converted back into a string. The decrypted rotational message is then further decrypted using the Hill cipher technique to eventually expose the original message once the inverse of the key matrix $(PK_1)^{-1}$ is calculated as de-scribed in Algorithm 2.

Algorithm 2: Extraction and decryption

1. Get the stego video frame
2. Extract the hidden rotated encrypted message
3. Use the same second PK_2 as in the encryption phase
4. Divide the rotated encrypted message into number of blocks equal to the PK_2 length(L)
5. Compute $RLD_d = (2 * L) * 4 - RLD_e$
6. For all rotated encrypted message blocks, do
   a. Transform each rotated encrypted block to binary
   b. Reshape the binary digits into one row matrix
   c. Rotate the one row matrix RLD_d times to the left
   d. Transform the resulting matrix to a string
   e. Get the decrypted rotated matrix
   End for
7. Get the key matrix inverse $(PK_1)^{-1}$
8. Decrypt the decrypted rotated message using Hill cipher decryption technique
9. Get the original message

4.2.1 Encryption and Hiding Flowchart

Encryption and Hiding process is divided into two types of encryptions: Hill cipher and Rotation left digits encryption, then the embedded DWT is applied in the selected video frame. The first encryption process is represented in the Hill cipher technique, where the secret text message is divided into several fixed blocks, and each block is multiplied by the key matrix (PK_1) to generate the encrypted Hill message. The second encryption process is represented in the rotation left digits technique; the second private key (PK_2) is generated from the resized pixel video frame, which is resized to one row and multiple columns equal to the encrypted Hill message length (L). Rotation is computed as $RLD_e = (L/2) * 4 + PK_2(I) + C$. The encrypted Hill message is embedded inside the video frame using the DWT technique as shown in Fig. 3. Then, the obtained stego video can be transmitted securely.

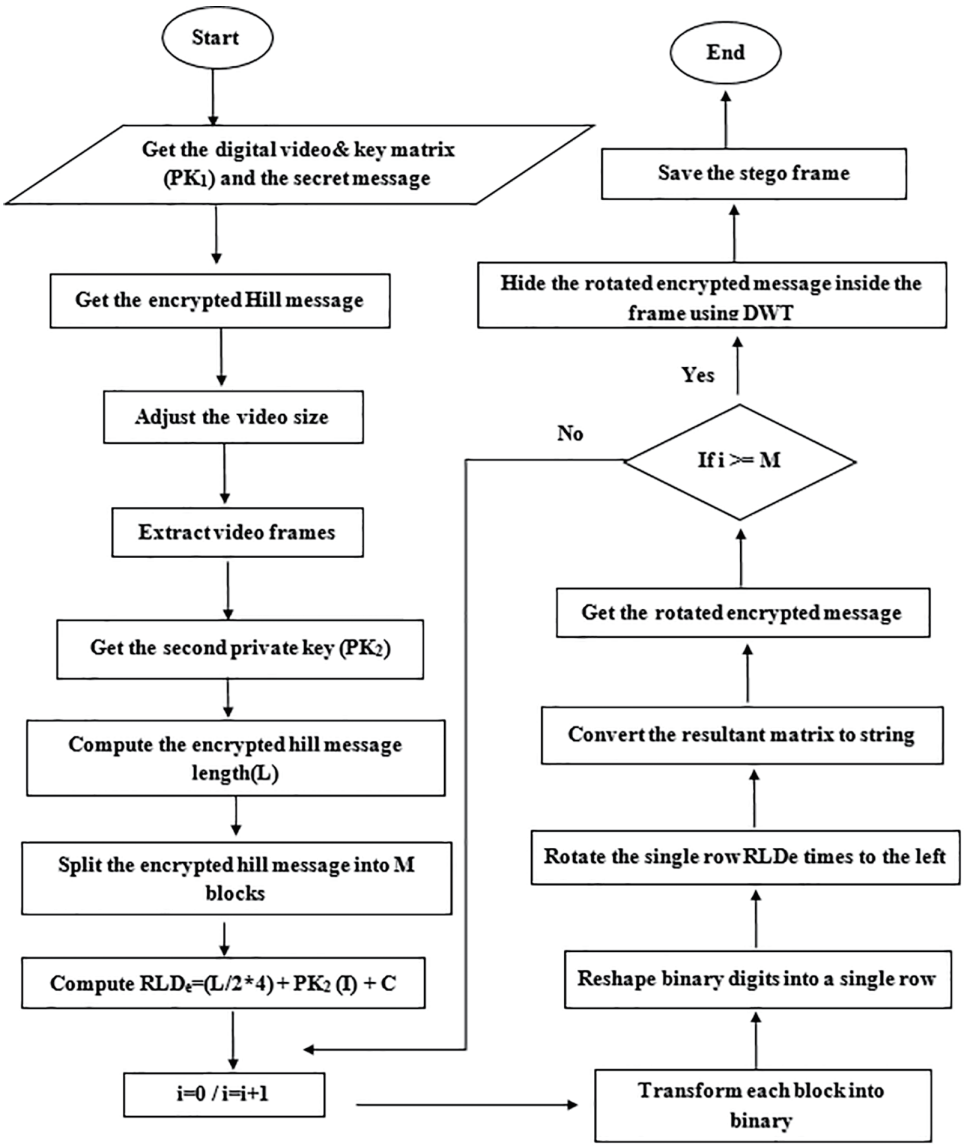

Figure 3: Encryption & hiding flowchart

4.2.2 Extraction and Decryption Flowchart

After stego frames transmission, the rotated encrypted message is extracted from the stego video frames using the DWT extraction process, then this rotated encrypted message will be decrypted first using $RLD_d = (L*2) * 4 - RLD_e$ and (PK_2) to obtain the encrypted Hill message. Then, the second decryption process will be applied with the inverse key matrix $(PK_1)^{-1}$ and encrypted Hill message, which will obtain the original text message as shown in Fig. 4.

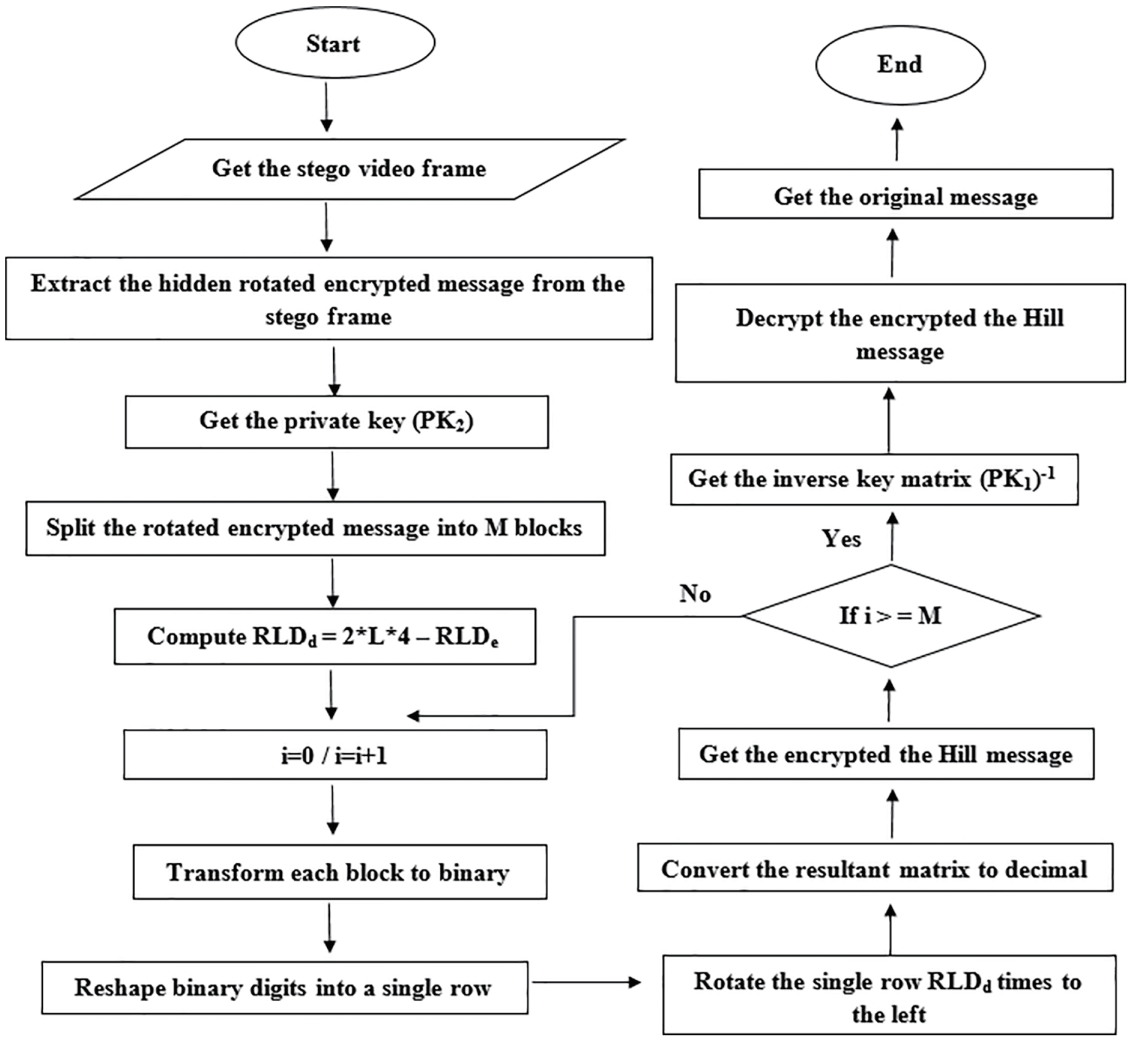

Figure 4: Extraction & decryption flowchart

5 Experimental Results

This section presents the findings from the implementation of the proposed lightweight secret message protection model hiding inside video frames, followed by a detailed discussion to interpret and analyze the results. The proposed method was applied with R2024a MATLAB. The proposed method was applied to different frame dimensions and different message sizes and tested with MSE, PSNR, and CC evaluation metrics. The MSE is calculated as follows:

$$MSE_x = \frac{1}{N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[S(i,j) - R(i,j)]^2, \tag{4}$$

where N equals m $*$ n. m is the number of rows, and n is the number of columns in the frame. The encrypted message value is S, and R represents the original secret message in decimal. The PSNR is defined as follows:

$$PSNR = 10*\log_{10}*\frac{(MAX_I)^2}{MSE_t}, \tag{5}$$

where MAX is the maximum pixel value in a certain video frame, which equals 255. The CC is computed as follows:

$$CC = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 (y_i - \overline{y})^2}}, \tag{6}$$

where x_i is the original message value, $\overline{x}$ is the mean of x, y_i is the value of the encrypted message and $\overline{y}$ is the mean of y. The SSIM is computed as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{7}$$

where μ_x, μ_y are the mean of the two frames. σ_x^2, σ_y^2 are the variances. σ_{xy} is the covariance. C_1, C_2 are small constants for stability. The MAE is computed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{3} |x_{c,i} - y_{c,i}|, \tag{8}$$

where N is the total number of frame pixels, c is the RGB color channels. The NPCR is computed as:

$$NPCR = \left(\frac{C}{N}\right) \times 100, \tag{9}$$

where C is the number of changed pixels, N is the total number of pixels. The UACI is computed as:

$$UACI = \frac{1}{N}\sum_{i=1}^{N} \frac{|x_i - y_i|}{255} \times 100, \tag{10}$$

where x_i, y_i are the pixel values across all the channels, $N = \mathrm{m} * \mathrm{n}$ where m is the total number of rows and n is the columns number in the frame.

The result must be very high in MSE and low in PSNR and CC according to the comparison between the original and encrypted message while in the decryption phase, the result must be: MSE = 0, PSNR = ∞ and CC = 1 between the original and decrypted messages and the same between the original and stego video frames, this indicates they are the same whether the messages or frames. Also, SSIM must be near 1, MAE must be near 0, NPCR must be near 0% and UACI must be near 0, which indicates that the two frames are identical to achieve a high security level.

The key generation, encryption, and decryption times for different video sizes were measured as presented in Table 4, where the used message size ranges from 10 bytes to 300 bytes in this approach.

The lightweight model consistently showed lower encryption and decryption times compared to other related encryption methods. The second private (PK_2) key is generated from video frame color values (RGB), and the secret message is divided into M blocks based on the private key size to do $RLD_{e,d}$ operations, taking into consideration the following issues:

- Changing the private key value in either the encryption or decryption phase will affect the original message.
- Choosing R or G, or B as pixel values will affect the evaluation of metrics of MSE, PSNR, and CC. If one of them can't do the encryption process in a perfect way, one of the other two will do it perfectly.
- Encryption and decryption times are not constant, so it doesn't depend on the message size but differs according to the private key results from R or G, or B channels, as shown in Fig. 5.

Table 4: Second Private Key (PK_2) generation, encryption and decryption time

Message size (Bytes)	Key generation time (s)	Encryption time (s)	Decryption time (s)
10	0.0073	0.00083	0.00073
20	0.0019	0.00081	0.00075
30	0.0023	0.0008	0.00073
50	0.0023	0.0006	0.00081
70	0.0027	0.00071	0.00073
80	0.0017	0.00082	0.00083
100	0.0018	0.00072	0.00097
150	0.0019	0.00088	0.00088
180	0.0028	0.00084	0.00099
200	0.0016	0.00083	0.00098
250	0.0036	0.0014	0.0014
300	0.0018	0.00096	0.00096

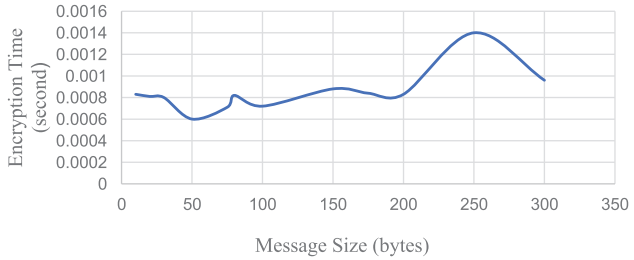

Figure 5: Message length vs. encryption time

6 Discussion

A triple lightweight model is applied and demonstrates a clear advantage in terms of computational efficiency. The overall encryption and decryption processes were faster and required less computational power, as indicated in Table 4. This is especially beneficial for real-time video streaming services or low-resource devices (e.g., mobile phones or embedded systems). While the presented model reduced computational and memory overheads significantly, it also maintained a high level of security by measuring the MSE, PSNR, and CC for different video sizes and frame dimensions as indicated in Table 5.

Table 5: Original and encrypted messages blocks MSE, PSNR, and CC values for different MPEG video sizes, frame dimensions

Block size	Frame number	Video size: 7.43 MB			Video size: 14 MB			Video size: 43.8 MB		
		Frame dimensions [240, 320]			Frame dimensions [360, 480]			Frame dimensions [600, 1050]		
		Duration 1:57 s			Duration 1:57 s			Duration 1:57 s		
		MSE	PSNR	CC	MSE	PSNR	MSE	MSE	PSNR	MSE
10	50	169.12	25.85	0.23	3721.02	12.42	−0.13	169.01	25.85	0.23
20	220	2916.05	13.48	0.35	9409.04	8.4	0.38	9025.01	8.58	−0.19
30	390	7396.01	9.44	−0.09	7396	9.44	−0.09	5329.01	10.86	0.05
50	450	2704.14	13.81	−0.12	2704	13.81	−0.12	576.02	20.53	0.13
70	530	7396.22	9.44	−0.12	7396.1	9.44	−0.12	7396	9.44	−0.19

(Continued)

Table 5 (continued)

Block size	Frame number	Video size: 7.43 MB			Video size: 14 MB			Video size: 43.8 MB		
		Frame dimensions [240, 320]			Frame dimensions [360, 480]			Frame dimensions [600, 1050]		
		Duration 1:57 s			Duration 1:57 s			Duration 1:57 s		
		MSE	PSNR	CC	MSE	PSNR	MSE	MSE	PSNR	MSE
80	560	12544.12	7.15	−0.22	12544.05	7.15	−0.22	7225	9.54	−0.19
100	630	841.1	18.88	−0.05	841.04	18.88	−0.05	19881	5.15	−0.05
150	1050	6084.12	10.29	−0.07	9404.06	8.4	0.01	1521.01	16.31	−0.01
180	1400	6241.03	10.18	−0.01	1089.01	17.76	0.01	361	22.56	−0.01
200	1820	3025.15	13.32	−0.04	64.05	30.07	−0.04	640.01	30.07	−0.04
250	2200	6561.11	9.96	0.14	6561.05	9.96	0.14	7921	9.14	0.05
300	2890	10000.07	8.13	−0.03	1000.03	8.13	−0.03	4356	11.74	−0.06

The combination of HCT, RLD encryption, and DWT hiding technique proved to be robust against both computational and cryptanalytic attacks. The balance between strong content protection and minimal resource usage makes this model ideal for applications where lightweight security is required. Proposed work comparison is done using the 300 message blocks (the maximum secret message size used in this approach) to obtain more accurate results, as presented in Table 6.

Table 6: Proposed work comparison according to encryption and decryption time

Research name	Video size MB	Encryption time (s)		Decryption time (s)	
		Ref. no.	Proposed	Ref. no.	Proposed
Reference [37]	3.06	0.17	0.004934	0.17	0.001253
Reference [38]	2.33	18.940	0.004946	18.940	0.002836
	4.21	19.508	0.005583	19.508	0.004796
Reference [39]	1.12	1.04	0.000798	1.02	0.000971
	0.67	0.74	0.000998	0.73	0.000898

The suggested method is also compared with other related research and obtained better results in the three evaluation metrics: MSE, PSNR, and CC in the case of different frame dimensions and frame numbers according to encrypted messages, as shown in Table 7.

Table 7: Proposed work comparison according to MSE, PSNR, and CC metrics values

Research number	Image dimensions	MSE	PSNR	CC
Reference [40]	360 * 480	8004	18.8	−0.0277
	600 * 1050	7551	20.81	−0.09
Reference [41]	360 * 480	6201	23.5005	0.1317
	600 * 1050	9677	18.972	−0.2448
Proposed	360 * 480	9404	8.58	−0.19
	600 * 1050	12544.05	7.15	−0.22

The same images and frames dimensions are compared as indicated in Table 8. The original video frame is identical to the stego frame according to the same pixel values of each frame, and the values of MSE, PSNR, and CC are 0 ∞ and 1, respectively, between the original frames and stego frames for the same message length as indicated in Table 9.

Table 8: Proposed work comparison according to the same images and frames, dimensions

Research name	Video name	Frame Dim.	Total frames	MSE		PSNR		CC	
				Ref. no.	Proposed	Ref. no.	Proposed	Ref. no.	Proposed
	Train	352 * 192	576	105.458	4408	38.015	13.29	−0.0006	−0.032
Reference [42]	Flamingo	352 * 192	464	105.74	3587.3	38.014	15.015	0.0014	−0.022
	Rhino	320 * 240	666	105.505	8167	38.014	9.55	0.0003	−0.07
	Xylophone	270 * 480	345	8687	11881	8.743	7.38	0.00019	−0.05
	Viptraffic	1440 * 1920	199	6907	12769	9.747	7.07	0.00046	−0.05
Reference [43]	Handshake-Right	1080 * 1920	179	10069	12996	8.102	6.99	0.00078	−0.02
	Single Ball	360 * 480	599	11803	11881	7.411	7.38	0.00123	−0.07
	Shuttle	720 * 1280	256	8146	11236	9.026	7.62	0.00130	−0.03

Table 9: Difference between original and stego video frames pixel values used in this approach

Message length (Bytes)	Original frame	Stego frame	MSE	PSNR	CC
10	[96 81 76 124 98 ...]	[96 81 76 124 98 ...]	0	∞	1
20	[117 145 162 159 156 156 ...]	[117 145 162 159 156 ...]	0	∞	1
30	[142 184 187 175 188 ...]	[142 184 187 175 188 ...]	0	∞	1
50	[126 104 103 120 . . .]	[126 104 103 120 . . .]	0	∞	1
70	[120 92 95 129 133 . . .]	[120 92 95 129 133 . . .]	0	∞	1
80	[138 135 135 125 . . .]	[138 135 135 125 . . .]	0	∞	1
100	[93 32 28 52 102 . . .]	[93 32 28 52 102 . . .]	0	∞	1
150	[142 137 171 176 . . .]	[142 137 171 176 . . .]	0	∞	1
180	[157 93 96 102 117 . . .]	[157 93 96 102 117 . . .]	0	∞	1
200	[162 157 177 166 ...]	[162 157 177 166 . . .]	0	∞	1
250	[121 133 157 169 ...]	[121 133 157 169 . . .]	0	∞	1
300	[222 219 211 210 196 . . .]	[222 219 211 210 196 . . .]	0	∞	1

Additionally, the SSIM, MAE, NPCR, and UACI indicate the identically of the original and stego video frames as shown in Table 10. RGB values impact on MSE, PSNR, and CC values, so the optimization of these metrics depends on the private key value as indicated in Table 11. The suggested approach can be expanded to use other codecs, including AVI, in addition to MPEG video frames, as shown in Table 12.

Table 10: SSIM, MAE, NPCR, and UACI values according to the Original and Stego Frame Comparison

RGB channel	Frame number	Block size	SSIM	MAE	NPCR	UACI
R	60	10	0.9984	0.0920	0.0304	0.0001
G	220	20	0.9918	0.1370	0.0463	0.0001
B	390	30	0.9963	0.1462	0.0336	0.0002
R	450	60	0.9984	0.1133	0.0449	0.0001
G	530	70	0.9980	0.1072	0.0554	0.0001
B	560	80	0.9973	0.1451	0.0469	0.0002
R	630	100	0.9981	0.0649	0.1098	0.0001
G	1050	150	0.9972	0.1585	0.0806	0.0002
B	1400	180	0.9962	0.1184	0.0719	0.0002
R	1820	200	0.9964	0.1714	0.1221	0.0002
G	2200	250	0.9960	0.1435	0.2530	0.0002
B	2890	300	0.9927	0.1919	0.1020	0.0002

Table 11: Second Private Key (Pk_2) Value Impact on MSE, PSNR, and CC metrics

Frame number	Message size	PK Generation			MSE			PSNR			CC		
		R	G	B	R	G	B	R	G	B	R	G	B
99	10	0.003	0.00267	0.00735	10609	64	4489	7.87	30.07	11.61	−0.22	−0.12	−0.2
215	20	0.00176	0.00215	0.00825	1024	2500	3969	18.03	14.15	12.14	0.13	−0.21	0.02
325	30	0.00156	0.00210	0.00177	10000	121	12544	8.13	27.3	7.15	−0.09	−0.23	−0.29
415	50	0.00187	0.00245	0.0017	7225	961.05	841	9.54	18.3	18.88	−0.01	0.06	−0.12
554	70	0.002	0.00143	0.0019	21609	31684	20736	4.78	3.12	4.96	0.03	0.09	−0.06
595	80	0.00164	0.00140	0.00142	10201	2704	4	8.04	13.81	42.11	0.05	0.07	0.03
655	100	0.00544	0.00457	0.00203	841.11	1024	961.16	18.88	18.03	18.3	0.08	−0.03	−0.05
706	150	0.00611	0.0020	0.00166	49.23	289	361	31.21	23.52	22.56	0.04	0.07	0.03
855	180	0.00216	0.0025	0.00667	0	2304.02	9	77.9	14.51	38.59	0.01	0.03	−0.04
1556	200	0.00349	0.00187	0.00198	7744	484.17	0	9.24	21.28	74.06	−0.04	−0.04	−0.04
1763	250	0.00167	0.0020	0.0089	1600	8464	0	16.09	8.86	72.37	0.1	0.01	−0.01
2480	300	0.00186	0.00202	0.0028	576	5929	11644	20.53	10.4	7.46	0.05	0.01	−0.04

Table 12: MSE, PSNR, and CC metrics values according to different AVI video sizes and frame dimensions

Message length (Bytes)	Video size: 7.93 MB			Video size: 76.3 MB			Video size: 31.8 MB		
	Frame dimensions [240, 320]			Frame dimensions [600, 1050]			Frame dimensions [360, 480]		
	MSE	PSNR	CC	MSE	PSNR	CC	MSE	PSNR	CC
10	14641	6.48	1	14641	6.48	1	12544	7.15	−0.17
20	484	21.28	0.38	4761	11.35	0.22	11449	7.54	−0.12
30	11025	7.71	−0.05	11025	7.71	−0.05	9216	8.49	−0.01
50	841	18.88	0.06	841	18.88	0.06	10000	8.13	0.07
70	1849	15.46	0.19	1849	15.46	0.19	14400	6.55	−0.27
80	1855	15.55	0.12	5329	10.86	0.13	10201	8.04	−0.34
100	7569	9.34	0.04	7569	9.34	0.04	2601	13.98	−0.16
150	12769	7.07	0.05	12769	7.07	0.05	4356	11.74	0.04
180	17956	5.59	0.01	17956	5.59	0.01	13924	6.69	0
200	9025	8.58	0.04	9025	8.58	0.04	2500	14.15	0.07

(Continued)

Table 12 (continued)

Message length (Bytes)	Video size: 7.93 MB			Video size: 76.3 MB			Video size: 31.8 MB		
	Frame dimensions [240, 320]			Frame dimensions [600, 1050]			Frame dimensions [360, 480]		
	MSE	PSNR	CC	MSE	PSNR	CC	MSE	PSNR	CC
250	13689	6.77	−0.04	13689	6.77	−0.04	9604	8.31	0.04
300	4356	11.74	0.01	4356	11.74	0.01	6400	10.07	0.03

7 Conclusion

This research has developed a robust triple lightweight model for concealing secret messages using HCT, RLD, and DWT processes. The proposed method employs two distinct private keys: the first is the Hill cipher key matrix (PK_1), and the second is generated from video frames to create the second key (PK_2). The secret message is initially encrypted with the Hill cipher, then encrypted again with the second key. This second encrypted message is resized into a one-row matrix with multiple columns, matching the size of the encrypted message, resulting in a rotated encrypted message that ranges from 10 to 300 bytes in this study. Finally, the Hill-encrypted secret message is further encrypted using RLD_e and embedded into the video frame using the DWT insertion technique. Once the encrypted rotated secret message has been transmitted, it is first retrieved and decrypted using the same (PK_2), RLD_d to obtain the encrypted Hill message. The original text message is then obtained through a second decryption using $(PK_1)^{-1}$. According to metrics like MSE, PSNR, and CC, the suggested approach demonstrated exceptional performance in terms of encryption and decryption execution times as well as exceptional resistance to a variety of attacks. The original and stego frames were compared using SSIM, MAE, NPCR, and UACI metrics, which showed that they are almost the same. Combining HCT and RLD has been demonstrated to provide more secure data transmission in video frames as opposed to pictures.

The lightweight architecture of the approach supports its usability, allowing for efficient execution even on systems with limited resources. Confidentiality is maintained using the Hill cipher and RLD techniques, which shield the hidden data from unwanted access, while integrity is maintained by employing DWT to embed the encrypted message in an organized manner. However, when working with high-resolution videos like HD or Ultra HD, the computational overhead from frame extraction and key generation can become a bottleneck, even with its promising speed. Future work could focus on optimizing the model for lower power consumption, improving its performance for ultra-high-definition (4K/8K) videos, and further enhancing resistance to advanced cryptographic attacks.

Acknowledgement: The authors would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Sara H. Elsayed, conceptualization, software, writing, and preparation; Rodaina Abdelsalam, data curation and investigation; Mahmoud A. Ismail Shoman, resources and project administration; Raed Alotaibi, writing and visualization; Omar Reyad, idea proposal, review and editing, and supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The video file that supports the findings of this research is publicly available as indicated in the references. Source code and proposed work data is available in [44].

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Hosny KM, Zaki MA, Lashin NA, Fouda M, Hamza HM. Multimedia security using encryption: a survey. IEEE Access. 2023;11:63027–56. doi:10.1109/ACCESS.2023.3287858.
2. Reyad O, Karar ME. Secure CT-image encryption for COVID-19 infections using HBBS-based multiple keystreams. Arab J Sci Eng. 2021;46(4):3581–93. doi:10.1007/s13369-020-05196-w.
3. Kadry H, Farouk A, Zanaty EA, Reyad O. Intrusion detection model using optimized quantum neural network and elliptical curve cryptography for data security. Alex Eng J. 2023;71(1):491–500. doi:10.1016/j.aej.2023.03.072.
4. Mehta K, Dhingra G, Mangrulkar R. Enhancing multimedia security using shortest weight first algorithm and symmetric cryptography. J Appl Secur Res. 2024;19(2):276–99. doi:10.1080/19361610.2022.2157193.
5. Reyad O, Mansour H, Zanaty E, Heshmat M. Medical image privacy enhancement using key extension of advanced encryption standard. Sohag J Sci. 2024;9(3):211–6. doi:10.21608/sjsci.2024.250176.1154.
6. Asghar A, Shifa A, Asghar MN. Survey on video security: examining threats, challenges, and future trends. Comput Mater Contin. 2024;80(3):3591–635. doi:10.32604/cmc.2024.05465.
7. Muhammed Z, Anwar Z, Javed AR, Saleem B, Abbas S, Gadekallu TR, et al. Smartphone security and privacy: a Survey on apts, sensor-based attacks, side-channel attacks, google play attacks, and defenses. Technologies. 2023;11(3):76. doi:10.3390/technologies11030076.
8. Kıran HE. A novel chaos-based encryption technique with parallel processing using CUDA for mobile powerful GPU control center. Chaos Fractals. 2024;1(1):6–18. doi:10.69882/adba.chf.2024072.
9. Alaca Y, Celik Y, Goel S. Anomaly detection in cyber security with graph-based LSTM in log analysis. Chaos Theory Appl. 2023;5(3):188–97. doi:10.51537/chaos.1348302.
10. Thomas G, André F, Matthias K. Forensic analysis of video file formats. Digit Investig. 2014;11:S68–76. doi:10.1016/j.diin.2014.03.009.
11. Potdar U, Talele KT, Gandhe ST. Comparison of MPEG video encryption algorithms. In: Proceedings of the International Conference on Advances in Computing, Communication and Control; 2009 Jan 23–24; Mumbai, India. New York, NY, USA: Association for Computing Machinery; 2009. p. 289–94. doi:10.1145/1523103.1523163.
12. Dalal M, Juneja M. A survey on information hiding using video steganography. Artif Intell Rev. 2021;54(8):5831–95. doi:10.1007/s10462-021-09968-0.
13. Azanuddin A, Kartadie R, Erwis F, Boy AF, Nasyuha AH. A combination of Hill cipher and RC4 methods for text security. Telecommun Comput Electron Control. 2024;22(2):351–61. doi:10.12928/telkomnika.v22i2.25628.
14. Hadi HHH, Neamah AA. An image encryption method based on modified elliptic curve Diffie-Hellman key exchange protocol and Hill Cipher. Open Eng. 2024;14(1):20220552. doi:10.1515/eng-2022-0552.
15. Al-Batah MS, Alzboon MS, Alzyoud M, Al-Shanableh N. Enhancing image cryptography performance with block left rotation operations. Appl Comput Intell Soft Comput. 2024;1(1):3641927. doi:10.1155/2024/3641927.
16. Wadho SA, Meghji AF, Yichiet A, Kumar R, Shaikh FB. Encryption techniques and algorithms to combat cybersecurity attacks: a review. Trans Comput Sci. 2023;11(1):295–305. doi:10.21015/vtcs.v11i1.1521.
17. Abubakar AM, Kauthar K. Hybrid of least significant bits and most significant bits for improving security and quality of digital image steganography. WSEAS Trans Comput. 2023;22:253–62. doi:10.37394/23205.2023.22.29.
18. Rabie T, Baziyad M. The pixogram: addressing high payload demands for video steganography. IEEE Access. 2019;7:21948–219. doi:10.1109/ACCESS.2019.2898838.
19. Suhail AT, Ayoub HG. A new method for hiding a secret file in several WAV files depends on circular secret key. Egypt Inform J. 2022;23(4):33–43. doi:10.1016/j.eij.2022.06.003.
20. Vergara GF, Giacomelli P, Serrano ALM, Mendonça FLLD, Rodrigues GAP, Bispo GD. Stego-STFAN: a novel neural network for video steganography. Computers. 2024;13(7):180. doi:10.3390/computers13070180.
21. Dutta S, Saini K. Securing data: a study on different transform domain techniques. WSEAS Trans Syst Control. 2021;16:23203–2021. doi:10.37394/23203.2021.16.8.
22. Oudah MK, Abed AN, Khudhair RS, Kaleefah SM. Improvement of image steganography using discrete wavelet transform. Eng Technol J. 2022;38(1):83–7. doi:10.30684/etj.v38i1A.266.

23. Durafe A, Patidar V. Development and analysis of IWT-SVD and DWT-SVD steganography using fractal cover. J King Saud Univ—Comput Inf Sci. 2022;34(7):4483–98. doi:10.1016/j.jksuci.2020.10.008.
24. Karim NA, Ali SA, Jawad MJ. A coverless image steganography based on robust image wavelet hashing. TELKOMNIKA (Telecommun Comput Electron Control). 2022;20(6):1317–25. doi:10.12928/telkomnika.v20i6.23596.
25. Murari TV, KC R, ME R. Selective encryption of video frames using the one-time random key algorithm and permutation techniques for secure transmission over the content delivery network. Multimed Tools Appl. 2024;83(35):82303–42. doi:10.1007/s11042-024-18613-1.
26. Mao X, Hu X, Peng W, Gan Z, Ying Q, Qian Z, et al. From covert hiding to visual editing: robust generative video steganography. In: Proceedings of the MM'24: The 32nd ACM International Conference on Multimedia; 2024 Oct 28–Nov 1; Melbourne, VIC, Australia. doi:10.1145/3664647.3681149.
27. Ibrahim A. Video steganography using least significant bit in frequency domain. Int J Intell Comput Inf Sci. 2016;16(1):89–98. doi:10.21608/ijicis.2016.10012.
28. Zareai D, Balafar M, FeiziDerakhshi M. EGPIECLMAC: efficient grayscale privacy image encryption with chaos logistics maps and Arnold Cat. Evol Syst. 2023;14(6):993–1023. doi:10.1007/s12530-022-09482-w.
29. Guan M, Yang X, Hu W. Chaotic image encryption algorithm using frequency-domain DNA encoding. IET Image Process. 2019;13(9):1535–9. doi:10.1049/iet-ipr.2019.0051 .
30. Ur Rehman A, Liao X, Ashraf R, Ullah S, Wang H. A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. Optik. 2018;159:348–67. doi:10.1016/j.ijleo.2018.01.064.
31. Niyat AY, Moattar MH, Torshiz MN. Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt Lasers Eng. 2017;90(9):225–37. doi:10.1016/j.optlaseng.2016.10.019.
32. Faragallah OS, El-Sayed HS, El-Shafai W. Efficient opto MVC/HEVC cybersecurity framework based on arnold map and discrete cosine transform. J Ambient Intell Humaniz Comput. 2023;14(3):1591–606. doi:10.1007/s12652-021-03382-8.
33. Alarifi A, Sankar S, Altameem T, Jithin KC, Amoon M, El-Shafai W, et al. A novel hybrid cryptosystem for secure streaming of high efficiency H.265 compressed videos in IoT multimedia applications. IEEE Access. 2020;8:128548–73. doi:10.1109/ACCESS.2020.3008644.
34. Salama WM, Elkamchouchi H, Saleh Y. New video encryption schemes based on chaotic maps. Let Image Process. 2019;14(2):1–16. doi:10.1049/iet-ipr.2018.5250.
35. Hosny KM, Zaki MA, Hamza HM, Fouda MM, Lashin NA. Privacy protection in surveillance videos using block scrambling-based encryption and DCNN-based face detection. IEEE Access. 2022;10:106750–69. doi:10.1109/ACCESS.2022.3211657.
36. Adeniyi AE, Misra S, Daniel E, Bokolo A Jr. Computational complexity of modified blowfish cryptographic algorithm on video data. Algorithms. 2022;15(10):373. doi:10.3390/a15100373.
37. Kumar V, Mali SS, Rajender G, Medikondu NR. A secure system for digital video applications using an intelligent crypto model. Multimed Tools Appl. 2024;83(6):16395–415. doi:10.1007/s11042-023-16223-x .
38. Al-Khafaji BJ, Rahma AMS. A modern encryption approach to improve video security as an advanced standard adopted. Ibn AL-Haitham J Pure Appl Sci. 2024;37(2):460–70. doi:10.30526/37.2.3907.
39. Ojo O, Kareem M, Oyinloye O, Ikumpayi O. Stegovideo: an efficient mechanism for securing video data using steganography and cryptography techniques. J Sci Technol. 2024;42(2):50–63. doi:10.4314/just.v42i2.4.
40. Abu-Faraj MA, Al-Hyari A, Aldebei K, Alqadi ZA, Al-Ahmad B. Rotation left digits to enhance the security level of message blocks cryptography. IEEE Access. 2022;10(12):69388–97. doi:10.1109/ACCESS.2022.3187317.
41. Alzyoud M, Saleh Alomar A, Al-shanableh N, Al-Batah M, Alqadi Z, Al-Hawary S, et al. The cryptography of secret messages using block rotation left operation. Appl Math Inf Sci. 2024;18(2):395–402. doi:10.18576/amis/180214.
42. Dhingra D, Dua M. A chaos-based novel approach to video encryption using dynamic S-box. Multimed Tools Appl. 2024;83(1):1693–723. doi:10.1007/s11042-023-15593-6.
43. Jaafar A, Mohammed AA. A colour video encryption scheme based on chaotic maps. Taiz Univ Res J. 2024;39(39).
44. Hamdy S. Lightweight-multi-layered-research [Internet]. [cited 2025 Jul 1]. Available from: https://github.com/shamdy1993/Lightweight-Multi-Layered-Research.