Research Article

Fadhil Abbas Fadhil, Farah Tawfiq Abdul Hussien Alhilo*, and Mohammed T. Abdulhadi

Enhancing data security using Laplacian of Gaussian and Chacha20 encryption algorithm

https://doi.org/10.1515/jisys-2024-0191
received March 22, 2024; accepted May 29, 2024

Abstract: Concealing sensitive information inside multimedia is very interesting in secure communication due to its wide application. This study discusses the different difficulties involved in embedding the ciphertext inside video frame without affecting the imperceptibility and the strength resistance against different cyberattacks. The main goal of this study is to suggest a novel technique to conceal encrypted data inside video frames securely. This is done by combining the Laplacian of Gaussian (LoG) edge detection algorithm and Chacha20 encryption algorithm. LoG facilitates to detect the suitable location inside video frame for concealing data. While ChaCha20 is used to encrypt data before embedding them inside these locations, which augmenting data security. The proposed method involves sequence of steps involving detecting edges inside video frames, determining the suitable edges for concealing data, encrypting data by XORing it with the encryption key which is generated using ChaCha20 algorithm, embedding the encrypted data inside the determined edges, and then reconstruct the video frame to rebuild the video that involves the concealed encrypted data and keeping the lowest level of visual distortion. The experimental results showed that combining these two approaches provide fast, robust, and secure method, which can be seen by evaluating the system using mean square error, peak signal-to-noise ratio, correlation, number of pixels change rate, unified average changing intensity, and entropy, these evaluation metrics provide excellent results. This study suggests a strong and novel method to embed the encrypted data inside video frames which can be employed in secure communication, copyright protection, and data authentication. Merging LoG and ChaCha20 algorithms produce perfect results in both security and visual perception quality, that provides a means for farther achievement in secure data embedding techniques.

Keywords: LoG, Chacha20, Gaussian blur, edge detection, text encryption

Abbreviations

LoG	Laplacian of Gaussian
PSNR	peak signal-to-noise ratio
MSE	mean square error
NPCR	number of pixels change rate
UACI	unified average changing intensity
Exp	exponential

* **Corresponding author: Farah Tawfiq Abdul Hussien Alhilo**, Department of Computer Sciences, University of Technology, Baghdad, 10066, Iraq, e-mail: farah.t.alhilo@uotechnology.edu.iq
Fadhil Abbas Fadhil: Department of Computer Sciences, University of Technology, Baghdad, 10066, Iraq, e-mail: Fadhil.a.fadhil @uotechnology.edu.iq
Mohammed T. Abdulhadi: Department of Computer Sciences, University of Technology, Baghdad, 10066, Iraq, e-mail: mohammed.t.abdulhadi@uotechnology.edu.iq

Nomenclatures

G	Gaussian kernel
X	X coordinate
Y	Y coordinate
Δ	Delta

Greek symbols

σ	Standard deviation
∂	Derivative
π	Semi-vertex angle of the conical nose (Figure 1), rad.

1 Introduction

Due to the continuous evolution of information security, protecting sensitive information is considered ultimate requirement in various applications. Using hybrid techniques to conceal encrypted data within

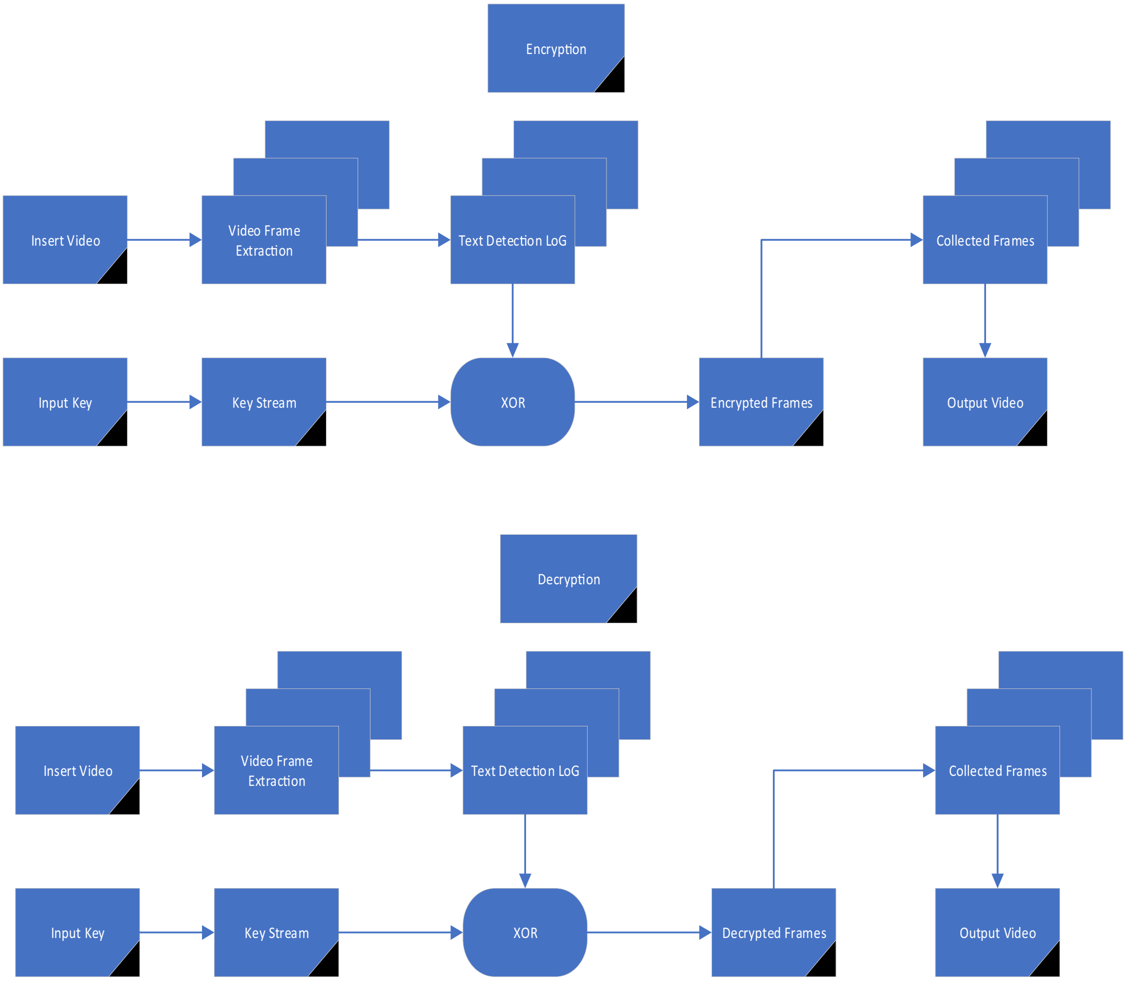

Figure 1: The proposed system.

multimedia environment is considered a powerful method to perform secure communication [1]. This study suggests to integrate two strong algorithms, Laplacian of Gaussian (LoG) edge detection technique and ChaCha20 encryption algorithm in order to select the appropriate locations for concealing the encrypted data inside video frames.

LoG is a powerful edge detection algorithm that support concealing information within perception visual content, while ChaCha20 encryption algorithm is used to strengthen concealing the encrypted data due to its speed and secrecy [2].

There are several issues and challenges that are addressed by the proposed approach which are discussed as follows. Many of the current edge detection techniques face difficulties in choosing the suitable edge accurately for concealing data within frames especially with complex video scene that may lead to visual distortion. In terms of security, using a weak encryption method or an inadequate key management system may expose sensitive data to different attacks. Further, achieving a balance between data concealment and perceptual impact such that the perceptual effect is not noticeable is difficult, especially with the huge amount of data to be concealed. High complexity represents an important challenge for real-world applications, where time is considered an important factor; it is also a difficulty for limited resource systems.

The proposed system addresses the discussed challenges through several means. Using the LoG algorithm augments the accuracy in selecting the suitable edge location for embedding data that prevents noise with high reliability and reduces the inappropriate selection of the edge location. Further, using the ChaCha20 encryption algorithm supports concealed data security due to its features of speed, high performance, and immunity to different types of well-known attacks. The proposed system guarantees imperceptibility due to the careful selection of the concealing location to reduce virtual distortion by integrating edge detection and encryption.

The LoG method represents the backbone in the proposed system because it offers strong approach to combine encrypted data in a visual media. As a result, the concealed data remain unseen to human eye and keep the video content uncorrupted.

The integration between LoG and ChaCha20 techniques support the strength for both of them, as well as it is a new era for securing sensitive information using video frames. The purpose of Merging these methods together is to keep secret information immune against intruders and attacks and securing protecting information during storage and transmission.

The suggested combination of these methods can be considered a novel method for a secure encryption by concealing the ciphertext within digital video. This hybrid method can be employed in different applications like secure end to end communication, video conference, and surveillance.

The aim of this study is to combine strong algorithms to develop a powerful novel method for embedding encrypted data within video frames, which support resistance against unauthorised access. The proposed approach aims to enhance the accuracy of selecting a suitable location for embedding the encrypted data with the least visual distortion and ensure the security of the encrypted data, by concealing them inside the video frame to increase the difficulty in detecting them. An additional objective is to maintain the quality of visual perception of the video frames so that the embedding process remains unpredictable for intruders. The following are the main contributions of this study:

- Protecting important details and reducing the visual effects due to the concealment of the encrypted text inside the chosen locations of video frames.
- Combining LoG edge detection and ChaCha20 algorithms to increase security. It provides multilevel security, especially, ChaCha20 is a fast and strong defence against many cybersecurity attacks.
- Providing a balance between the security level and efficient visual quality.

Concerning the remaining sections of the study, Section 2 presents related works, Section 3 explains the methodology, Section 4 provides the experimental results, Section 5 discusses comparison analysis, Section 6 discusses limitations and Section 7 presents the conclusion.

2 Related works

This section discusses the previous works that used different edge detection techniques in diverse applications.

One of these is the Edge Adaptive based on the Least-Significant-bit Matched Revisited (EALSBMR) approach [3]. The method detects edges using image processing and develops a comparison using MATLAB software. To identify edges, the Sobel approach is used on cover images. Then, secret bits are embedded by taking advantage of sharper edges. In this research, the EALSBMR method is used with Sobel operator assistance to achieve an application that is both faster and more reliable than the EALSBMR technique, with a reasonable compression ratio. To get edges, cover images are subjected to the Sobel edge detection approach. Subsequently, sharper edges are used to insert hidden information. This method offers a decent compromise between image quality and security. On the other hand, the strength of the method against different types of attacks is not measured accurately.

Two basic edge detection limits are presented in another study: edge thickness and edge connectivity [4]. A reliable edge detection technique using multiple threshold approaches (B-Edge) is suggested. The commonly used canny edge detector observes some gaps in optimal results and concentrates on selecting two thresholds. This approach chooses the simulated triple thresholds that target the key issues with edge detection: picture contrast, choosing effective edge pixels, solving mistakes, and resemblance to the ground truth. This closes the loopholes of the cutting-edge operator. The suggested method aims to improve both coloured and grayscale photos. However, the suitability of the method for different applications in real world is not considered.

Using the multi-criteria decision-making approach and visual cues from satellite pictures, Bausys et al. [5] suggested an approach involving the adaptive selection of edge detection techniques. Selecting the best technique for the selected satellite photos is not easy because no single right algorithm works for all situations. Instead, it depends on a variety of elements, including the raster image's content and acquisition, real-world image visual characteristics, and human vision. This method suffers from limited exploration in different environments or imaging models.

Xui and Ge [6] suggest a deep learning approach for image smoothing and edge identification, depending on the convolutional neural network. The findings demonstrate that the research strategy suggested in this study greatly enhances the edge effect, preserves the usefulness of edge information, and effectively addresses the issues of detecting edges and graping information. Simultaneously, it enhances the smoothing effect of the image and lowers the signal-to-noise ratio (SNR) of the smoothed image. Evaluating the method's performance and its robustness *vis-a-vis* other techniques for complex structured images is not considered in this study.

Xiaofeng et al. [7] suggested a detecting edges approach based on deep learning for cancer images, since the current edge detection for medical images is low in both image restoration accuracy and fitness of the optimisation coefficient, leading to detection outcomes with low information recall, bad smoothness, and weak detection accuracy. First, a three-dimensional model of the cancer image's surface structure reconstruction is created. Second, the fine-grained characteristics of the cancer cells in the cancer image are extracted using edge contour feature extraction approach. Ultimately, the multi-dimensional pixel feature distributed reconstruction model of the cancer image is built. The ultra-fine particle feature is recovered, and regional fusion and information reconstruction are realised by using the fine-grained feature segmentation approach. A combination deep learning system enables the adaptive optimisation of edge detection. Evaluating the performance of the method for different and complex structure images is not explored in this study.

Academics working on different applications currently find it difficult to identify underwater images, including tracking fish species, monitoring coral reef species, and counting marine life [8]. As underwater images are often subjected to distortion and light attenuation, pre-processing techniques are necessary to improve their value. The edges of the underwater photos were identified in this article through a variety of edge detection methods. A variety of specialised approaches, including thresholding, median filtering, Wiener filtering, and enhancement processing, are used to pre-process the images. For evaluating the effectiveness of each edge detection technique employed in the testing, coral reef photos are used as a dataset of underwater images. An underwater GoPro camera is used to take pictures of every coral reef dataset. However, evaluating the performance of the method in different environments and with different underwater images is not addressed.

In remote sensing processing, edges are discrete geometric features essential for higher-level object detection and recognition [9]. However, because of the significant speckle noise that causes false positives (type I errors), edge detectors intended for optical images typically perform poorly on SAR images. Consequently, numerous researchers have proposed edge detectors specially designed to address the features of SAR images. While these edge detectors may yield good results in independent assessments, the comparisons typically include a rather small set of (simulated) SAR images. Due to this, the generalised performance of the suggested approaches is not accurately reflected because real-world patterns are far more varied and complex. This leads to another issue, which is the absence of a quantitative standard in the industry. Therefore, it is currently impossible to compare fairly any edge recognition technique for SAR images. Therefore, this study aims to fill up such gaps by offering a thorough experimental assessment of edge recognition in SAR images. To achieve this, a benchmark is presented on SAR picture edge recognition techniques, by assessing several publicly available techniques, including those considered the cutting edge. The limited exploration of the method's performance with noise is a gap.

Arulananth et al. [10] presented the fast Pixel-based matching and contour mapping approach, which uses non-local and mask-propagation approaches for edge comparison of the reference and targeted frames. Since our approach uses information from both the first and previous frames, it can handle obstacles and resist noticeable item's visual fluctuation. Section 4 discusses the improvement in the performance of the suggested system, with tabulated and drawn data. The most notable reinforcements are found in detection probability and detection time. There is no comparison with other methods or any evaluation with complex images.

Sultana et al. [11] proposed a hybrid image steganography method of secret hiding by combining the edges and local binary pattern (LBP) code qualities. Only edge pixels that affect how successfully a new methodology conceals data are used in this strategy. A logical OR operation is used to apply multiple edge detectors and hybridise them to enhance the number of computed edge pixels. For this, a morphological dilatation process in the hybridised edge picture is used. For edge pixels, least significant bits (LSB) and all LBP codes are computed. These LBP codes, LSBs, and secret bits are then combined using an XOR operation. The LSBs of the edge pixels receive these inserted bits consequently. The limited exploration of the method's strength against different attacks and its performance in concealing different types of data inside images are not measured.

Sultana et al. [12] used additional edge pixels by employing edge detectors in the prediction error space (PES). The PES is generated by using a predictor on the cover image and then computing the prediction errors. The edge detector is used to mark the edges in PES. More information is acquired by the edge-error corresponding pixels than by the pertinent pixels, which does not result in an edge-error. To generate more edges and a better embedding capacity, the output from many edge detectors is also aggregated. In non-edge pixels, where $x > y$, x number of hidden bits and y number of bits are implanted. According to the simulation results, the suggested scheme performs better than its competitors on every performance-measuring criterion, such as payload, stego picture quality, and assault resistance. Its scalability for concealing different data applications in diverse images is not explored.

3 Methodology

This section discusses the algorithms employed to design the proposed system.

3.1 LoG edge detector

The Laplace response is paired with a Gaussian filter in the LoG, which is an extension of the Laplacian filter. Though it is more susceptible to noise than the first derivative variations, the Laplacian is an excellent tool for spotting thin edges [13]. Before using a Laplacian convolution to find edges in an image, a Gaussian filter is applied to reduce the amount of false edge detections [14]. However, some of the sharp edges may be smoothed out, decreasing the accuracy of edge localisation. Therefore, attention should be paid when handling the smoothing parameter [15]. Finally, zero-crossing – the main component of this algorithm – achieves thresholding. To do this, equation (1) estimates the Gaussian kernel. Equation (2) then takes the Laplacian of the Gaussian equation, and equation (3) realises the LoG filter in the following manner [16]:

$$G(X, Y;\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-x^2 - y^2}{2\sigma^2}\right), \tag{1}$$

$$\Delta^2\,(G(X, Y;\sigma)) = \frac{\partial^2}{\partial x^2}G(X, Y;\sigma) + \frac{\partial^2}{\partial y^2}G(X, Y;\sigma), \tag{2}$$

$$\text{LoG} = \Delta^2(G(X, Y;\sigma)) = \frac{1}{\pi\sigma^4}\left(\frac{x^2 + y^2}{2\sigma^2} - 1\right)\exp\left(\frac{-x^2 - y^2}{2\sigma^2}\right), \tag{3}$$

where σ represents the standard deviation of the Gaussian.

The LoG algorithm is used due to its ability to detect the appropriate location for embedding accurately, with the least perceptible distortion. This is due to two reasons, first, it focuses on picking the locations with the high-intensity changes without obvious distortion in video frames. Second, it prevents noise due to the incorporation of Gaussian smoothing with Laplacian filtering, which efficiently decreases noise in video frames.

3.2 Chacha20 encryption algorithm

One of the best ways to secure data is through encryption, ChaCha20 algorithm is the most widely used encryption technique available today. It is a fast and safe encryption method that can be used for various applications [17]. ChaCha20 is considered as a symmetric ciphering algorithm that uses a 256-bit key for both ciphering and deciphering. Its construction resists known attacks such as differential and linear cryptanalysis and is intended to offer a speedy and secure solution [18,19]. Furthermore, because of its great parallelisability, multi-core CPUs and other high-performance computing systems can readily be used with it. As a stream cipher, ChaCha20 encrypts data continuously as opposed to fixed-size blocks. The ciphertext is created by XORing the pseudo-random bits in the continuous keystream that is generated with the plaintext data [20].

[21] Here are the fundamental steps in the ChaCha20 encryption process:

1. Creation of encryption key: An encryption key of 256-bit length is created by using user-supplied key and arbitrary 96-bit nonce.
2. Initialisation: The initial ciphering state is generated using both the key and the nonce.
3. Data encryption: Using the cipher’s state for encrypting each data block, the state is updated after each block processing.
4. Output: The plaintext and the encrypting data step’s outcome are XORed to create the final ciphertext.

There are a number of reasons for using the ChaCha20 algorithm, including speed, security, parallelism and easy implementation [22]

There are several uses for ChaCha20 encryption, including [23]:

1. Secure conversations: VPNs and secure messaging apps both use ChaCha20 to safeguard conversations between participants.
2. File encryption: This technique encrypts data that are sent over a network or kept on a device.
3. Internet of Things (IoT) security: ChaCha20 protects IoT devices, which frequently have constrained processing power and necessitate the use of simple encryption techniques.
4. Web security: It can be applied to protect HTTPS connections and other web traffic.

ChaCha20 algorithm involves the following steps:

1. Generating key: A secret key of about 128 or 256 bit of length is used to initialise encryption key stream. Then, a unique value known as nonce for each encryption key is used to prevent reusing the same key in different encryption processes.
2. Key expanding: The secret key and the nonce are extended to produce key block which is used to build subsequent key stream blocks. The expansion includes several rounds of mixing and permutation to produce strong key features like randomness and unpredictability.
3. Pseudorandom key generation (PRNG): The initial key block is used as PRNG to produce a key of random length.

The ChaCha20 encryption algorithm is used for many reasons. First, it is a high-performance algorithm that has been improved to be compatible with modern equipment, which supports the fast encryption and decryption processes. Another feature is parallelisation, which makes its employment possible with multi-core processors and hardware, thereby supporting scalability and speed. High resistance against known attacks, such as differential and linear cryptanalysis, making it a strong and reliable choice. Big keys and nonce space provide the broad scope of a viable mix of ciphering keys and nonces. It has high efficiency with low complexity and sustained time complexity; in other words, the execution time is kept constant, regardless of the amount of input data. All these features make the ChaCha20 the best choice for real-time applications that need speed, security, and high efficiency.

3.3 Proposed system algorithm

As mentioned, some significant texts may surface in video recordings that are confidential and should not be shown to the general public. A novel algorithm has been proposed that chooses which texts in the video should be encoded. The main purpose of this approach is to encrypt text images extracted straight from the input video frames. A video clip is the input (with hidden texts anticipated). The following diagram (Figure 1) handles the partially encrypted video. The suggested technique is explained by Algorithm 1.

Algorithm (1), Encryption phase

```
Input: encryption key, video, secret text
Output: encrypted video
Start
Step 1: load video
Step 2: frame extraction
Step 3: convert frames to grayscale
Step 4: apply LoG edge detection
Step 5: select appropriate locations
Step 6: Generate encryption key using ChaCha20
Step 7: XORed text and encryption key
Step 8: embed the encrypted text into the selected locations
Step 9: collapse the frames
Step 10: release the resulting video
End
```

The edge detection process involves several steps. First, the image is smoothened using Gaussian blurring to minimise noise and address the greatest intensity changes and then the Laplacian filtering is done to improve the edges by exploring intensity gradient changes. Then, a threshold value is determined to discriminate between noise and edges. The edges with a value greater than the determined threshold are kept for further manipulation, to eliminate weak edges that are not suitable for the concealing process. The characteristics considered for choosing the suitable edge for concealing are strength, length, connectivity, and orientation, to minimise noticeable distortion. Next the region of interest (ROI) is determined within the selected edge, where secret data are to be concealed. ROIs are chosen based on the length and volume of the concealed data. Finally, the chosen edge and ROIs are adjusted to guarantee the best performance with the least visual impact. It may involve selected edge fine-tuning, ROI size, and location adjustment and threshold parameters refinement.

The XOR operation plays a principal role in creating the ciphertext. It is employed between a byte of the plaintext and a byte of the encryption key. The same key is used in the encryption and decryption processes. The importance of the XOR operation is represented by its reversibility, randomness, unpredictability and computational efficiency. All these features support the speed, performance and security of the ChaCha20 algorithm.

The system parameters comprise

(1) Gaussian Blur of size 7 × 7,
(2) Standard deviation to decide the image blurring degree,
(3) Laplacian filter of size 7 × 7,
(4) Encryption key of 256-bit length,
(5) Nonce of 96-bit size, and
(6) The number of encryption rounds is 20.

We recently employed text encryption to encrypt some text, and the outcome was excellent. The suggested method was thoroughly evaluated using various input colour picture samples taken from the input video, and it was shown to be secure and reliable, encrypting data is a wonderful way to protect it and ensure access when needed. The stages involved in encryption are as follows.

3.4 Text detection

The final section of this study explains how the LoG edge detector works and how it was used to find the writings in the movie. Of course, this process involves several mathematical calculations in the background. To explain everything, though, could be a bit intimidating. Consequently, the first three stages are indicated, describing the ideas underlying its functioning.

Sliding a fixed-size window across the image yields the biggest intensity fluctuation while moving in the X and Y axes. For every window found, a score R is determined. The necessary edges are then selected by applying a threshold to this score. For a flat area, R is frequently perceived as being small. A region is considered to be an edge if R is small, and an edge if R is large (Figure 2).

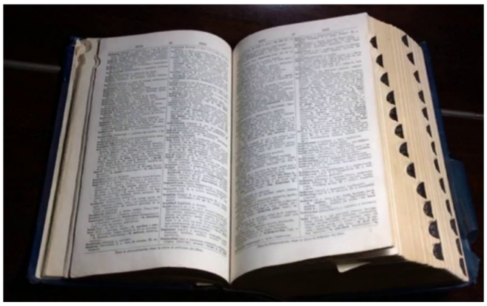

Figure 2: Frame represents book.

3.5 Keystream generation

It is widely recognised in cryptology that the key, the method of generation, and its unpredictability degree are significant variables in deciding how powerful the method is. Cause randomness tests are required for checking the resulting key. It is therefore difficult to hack. The key creation procedure is employed in the ChaCha20 encryption method.

3.5.1 Text encryption

The points are identified for encryption purposes by XORing with the keys after edge detecting, which are represented by the texts for the individual frames (Figure 3).

4 Results and discussion

Encryption is the best security technique for protecting digital video footage. Consequently, there are now two ways to encrypt digital video files: completely or partially. The key is essential to the ciphering operation because it enhances the approach, increases security, and makes it harder for intruders to crack. Three video clips were used to determine the measurement results using the suggested algorithm. The ciphering and deciphering results for the three selected samples used in the research findings are explained in Table 1. The suggested encryption methods have shown promising results concerning both time and the encryption process. Table 1 uses three different-sized films to illustrate these values (Figure 4).

Table 1: Time consumed for enciphering and deciphering for different file sizes

File name	Size	Type	#Frames	Frame sequence	Ciphering time	Average	Deciphering time	Average
The book	2.09 MB	MP4	138	1	2.8588258	2.1801140	3.3665222	2.6378514
				2	4.2349038		3.0847067	
				3	2.8358225		3.3364475	
				4	3.2516002		4.8714071	
				5	3.2013587		2.8763858	
				..	..		..	
				138	2.1923940		2.7389886	

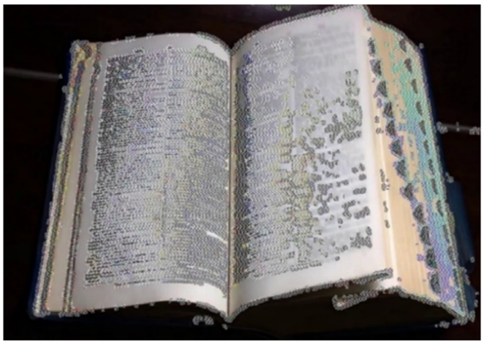

Figure 3: Detecting text in book frame.

Other measurement approaches were used to more accurately and objectively determine the amount of encryption in video encrypting, as human analysis is not the only way therefor. Numerous quality criteria exist. Table 2 displays the most significant ones (unified average changing intensity (UACI), peak signal-to-noise ratio (PSNR), number of pixels change rate (NPCR), mean square error (MSE), correlation, and entropy) used to measure the suggested technique. These criteria are numbered as follows: (1 = MSE, 2 = PSNR, 3 = Correlation, 4 = NPCR, 5 = UACI, and 6 = Entropy).

Table 2: Fidelity criteria

Video file name	Size	Criteria	Encryption with ChaCha20	Encryption with ChaCha20 and LoG edge detector
The book	2.09 MB (2,195,456 bytes)	1	26328.318	38.3324
		2	3.926	30.2900
		3	−0.0018	0.6721
		4	99.60	42.1161
		5	27.77	11.0516
		6	7.2571 Encrypt 7.7065 Decrypt	7.3523 Encrypt 7.1666 Decrypt

Table 2 indicates that the proposed method was both effective and resistant to attacks, as evidenced by the high MSE results and low PSNR values found. The recommended algorithm's correlation values were very good. The algorithm's strong defence against differential attacks was illustrated by the NPCR and UACI figures. The outcomes of the suggested method are excellent because they are close to the ideal value of entropy, which is known to be 88. This indicates that the recommended approach is successful in distributing the ciphered frame's pixels randomly. Importantly, the outcomes of the suggested system are good because they are close to the ideal value.

5 Comparison analysis

This section compares the proposed system with the related works mentioned in Section 2, involving several features as described in Table 3.

6 Limitations

The proposed system has several limitations, as indicated below:

- The main dependence is on edge detection and selection of the suitable edges. If the choice is not appropriate then there will be a noticeable distortion that reveals the embedded data
- There could be a variance in the performance of the method depending on the used video type. If the video contains complex contents or there are more changes in the visual elements, it represents a challenge that may reduce the method's efficiency or increase the visual distortion.
- The limited embedding capacity in cases where the video contains no suitable edges to conceal the encrypted data makes embedding of large encrypted data more difficult and leads to noticeable distortion in the visual contents.

Table 3: Comparative analysis

Paper	Storage capacity	Complexity	Speed	Randomness	Security	Suitability for real-world applications
3	Adequate	High	Differs according to the algorithm and hardware type	High	Low	Steganography
4	According to the frame number	Adequate	Adequate	Adequate	Not considered	Image analysis
5	Adequate	Adequate	High	Low	Not considered	Analysing images for satellite
6	High	According to computer vision, sensor fusion	High	Low	Not considered	Image processing
7	Adequate	High	Changes according to image size and hardware	Adequate	Not considered	Medical image
8	Low	Adequate	Adequate	Not considered	Not considered	Analysing the images underwater
9	According to the edge recognition techniques	Adequate	Low	According to edge recognition techniques	Not considered	Remote sensing
10	Low	Adequate	Adequate	Adequate	Not considered	In general, edge detecting
11	High	Adequate	High	High	Adequate	Steganography
12	Adequate	Adequate	Adequate	Not considered	High	Steganography
Proposed method	According to the text length	High	High	High	High	Data encryption, security applications, video conference, and surveillance

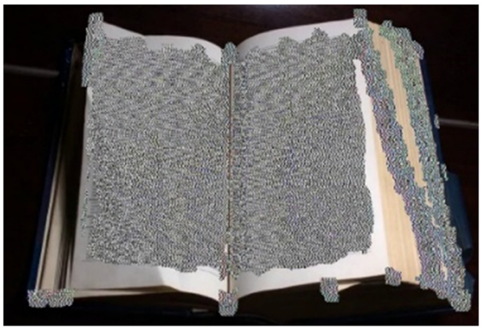

Figure 4: Encryption outcome of Figure 3.

7 Conclusion and future work

This study presents a novel approach for concealing encrypted data inside video frames by combining two strong approaches, LoG edge detection algorithm and ChaCha20 encryption algorithm. This is done by selecting the concealing locations by means of LoG edge detection algorithm and encrypting data using the ChaCha20 encryption algorithm. This combination offers a high level of security without affecting the visual quality. This is due to LoG ability to dedicate locations with high edge information to conceal with. On the other hand, the ChaCha20 encryption algorithm is a lightweight, fast, and strong encryption method which has immunity against many attacks that provide high security for the suggested method. This combination resulted in reducing the distortion in the visual contents and provide strong resistance against many attacks. This study provides a strong and secure method for many applications in the real world such as video surveillance, copy right production, and video conference, i.e. it provides a very secure and efficient method for video communication. Experimental results depending on metrics such as UACI, PSNR, NPCR, MSE, correlation, and entropy, prove the efficiency of this method.

There are several suggestions for future work, including an error correction code with the encrypted text before embedding inside the video, to reduce the impact of noise and error during embedding and transformation. Further, building a strong key management system that involves key generation, exchange, and storage, which involve key rotation and key exchange approaches, will support the encryption method and increase security.

Funding information: This work was not funded.

Author contributions: Farah Tawfiq Abdul Hussien Alhilo: conceptualisation, methodology, formal analysis, writing – original draft, and writing – review and editing. Fadhil Abbas Fadhil and Mohammed T. Abdulhadi: conceptualisation and methodology.

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this article.

Data availability statement: The data used to support the findings of this study are included within the article.

References

[1] Fadhil AF, Ali M, Safiullin N. The study on usage of table functions instead of basic operators inside encryption algorithm. Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). Yekaterinburg, Russian Federation; 2022. p. 320–3. doi: 10.1109/USBEREIT56278.2022.9923412.

[2] Hussien FTA, Rahma AMS, Wahab HBA. A Secure E-commerce environment using multi-agent system. Intell Autom Soft Comput. 2022;34(1):499–514. doi: 10.32604/iasc.2022.025091.

[3] Smitha GL, Baburaj E. Sobel edge detection technique implementation for image steganography analysis. Biomed Res Spec Issue. 2018;487–93. doi: 10.4066/biomedicalresearch.29-17-1212.

[4] Mittal M, Verma A, Kaur I, Kaur B, Sharma M, Goyal LM, et al. An efficient edge detection approach to provide better edge connectivity for image analysis. IEEE Access. 2019;7:33240–55. doi: 10.1109/ACCESS.2019.2902579.

[5] Bausys R, Kazakeviciute-Januskeviciene G, Cavallaro F, Usovaite A. Algorithm selection for edge detection in satellite images by neutrosophic WASPAS method. Sustainability. 2020;12(548):1–24. doi: 10.3390/su12020548.

[6] Xui H, Ge D. A novel image edge smoothing method based on convolutional neural network. Int J Adv Robotic Syst. 2020;17(3):1–11. doi: 10.1177/1729881420921676.

[7] Xiaofeng L, Hongshuang J, Yanwei W. Edge detection algorithm of cancer image based on deep learning. Bioengineered. 2020;11(1):693–707. doi: 10.1080/21655979.2020.1778913.

[8] Awalludin EA, Arsad TNT, Hj WNJ, Yussof W, Bachok Z, Hitam MS. A comparative study of various edge detection techniques for underwater images. J Telecommun Inf Technol. 2022;1:23–33. doi: 10.26636/jtit.2022.155921.

[9] Meester MJ, Baslamisli AS. SAR image edge detection: review and benchmark experiments. Int J Remote Sens. 2022;43(14):5372–438. doi: 10.1080/01431161.2022.2131480.

[10] Arulananth TS, Chinnasamy P, Babu JC, Kiran A, Hemalatha J, Abbas M. Edge detection using fast pixel based matching and contours mapping algorithms. Plos One. 2023;18(8):1–19. doi: 10.1371/journal.pone.0289823.

[11] Sultana H, Kamal AHM, Hossain G, Kabir MA. A novel hybrid edge detection and LBP code-based robust image steganography method. Future Internet. 2023;15(108):1–22. doi: 10.3390/fi15030108.

[12] Sultana H, Kamal AHM, Apon TS, Alam MGR. Increasing embedding capacity of stego images by exploiting edge pixels in prediction error space. Cyber Security Appl. 2024;2(100028):1–18. doi: 10.1016/j.csa.2023.100028.

[13] Hussien FTA, Rahma AMS, Wahab HBA. A block cipher algorithm based on magic square for secure e-bank systems. Comput Mater Continua. 2022;73(1):1329–46. doi: 10.32604/cmc.2022.027582.

[14] Fadhil FA, Hussien FTA, Khairi TWA, Nikolai Safiullin N. A proposed text encryption inside video using Harris corner detection and Salsa20 encryption algorithm. Baghdad Sci J, Online-First. 2024;7:1–15. doi: 10.21123/bsj.2023.9168.

[15] Hussien FTA, Khairi TWA. Performance evaluation of AES, ECC and logistic chaotic map algorithms in image encryption. Int J Interact Mob Technol. 2023;17(10):193–211. doi: 10.3991/ijim.v17i10.38787.

[16] Abdulhadi MT, Abbas AR. Human action behavior recognition in still images with proposed frames selection using transfer learning. Int J Online Biomed Eng. 2023;19(6):47–65. doi: 10.3991/ijoe.v19i06.38463.

[17] Karim AA, Nasser EF. Improvement of corner detection algorithms (Harris, FAST and SUSAN) based on reduction of features space and complexity time. Eng Technol J. 2017;35(Part B. 2):112–8. doi: 10.30684/etj.2017.138622.

[18] Hassan YA, Rahma AMS. Improving video watermarking through galois field GF(24) multiplication tables with diverse irreducible polynomials and adaptive techniques. Comput Mater Continua. 2024;78(1):1423–42. doi: 10.32604/cmc.2023.046149.

[19] Hasan IM, Ghani RF. Blockchain for authorized access of health insurance IoT system. IRAQI J Comput Commun Control Syst Eng. 2021;21(3):76–88. doi: 10.33103/uot.ijccce.21.3.7.

[20] Kamal ZA, Ghani RF. A proposed hash algorithm to use for blockchain base transaction flow system. Periodicals Eng Nat Sci. 2021;9(4):657–73. doi: 10.13140/RG.2.2.31831.14249.

[21] Ali YH, Ressan HA. Image encryption using block cipher based serpent algorithm. Eng Technol J. 2016;34(2):278–86. doi: 10.30684/etj.34.2B.10.

[22] Naser SM, Ali YH, Al-Jumeily D. Hybrid cyber-security model for attacks detection based on deep and machine learning. Int J Online Biomed Eng (iJOE). 2022;18(11):17–30. doi: 10.3991/ijoe.v18i11.33563.

[23] Hussien FTA, Rahma AMS, Wahab HBA. Design and implement a new secure prototype structure of e-commerce system. Int J Electr Comput Eng. 2022;12(1):560–71. doi: 10.11591/ijece.v12i1.pp560-571.