Engineering and Technology Journal
Journal homepage: engtechjournal.org

A Proposal Video Encryption Using Light Stream Algorithm

Ataa R. Alawi [a*], Nidaa F. Hassan [b]

[a] Computer Sciences, University of Technology, Bagdad-Iraq, ataarasol5@gmail.com

[b] Computer Sciences, University of Technology, Bagdad-Iraq, 110020@uotechnology.edu.iq

*Corresponding author.

Submitted: 20/04/2020 Accepted: 15/06/2020 Published: 25/03/2021

K E Y W O R D S

Video encryption, Video Security, ChaCha, Stream cipher, Features Detectors Operators.

A B S T R A C T

Video encrypting is one technique to protect digital videos, it used to avoid unwanted interference and viewing of the transmitted videos. In this paper, a new selective video cryptography algorithm is suggested using light stream algorithm. As it known video size is large in size and it consume time in the encryption process, ChaCha a light encryption algorithm has been used to reduce the encryption time, encryption is done by Xoring frames of video with the key generated from ChaCha algorithm, it produced an acceptable results from robustness point view, but still encryption process consumed time, thus to speed up this process, feature detection operator (FAST) is used to encrypt key points result from FAST operator, in addition key points from this is increased to optimized between speed and robustness of proposed algorithm. In evaluation process, some of measuring quality factors MSE, PSNR, Correlation, NPCR, UACI and entropy are specified for evaluating and comparing between two suggested encryption algorithms which gave good result in encryption process (ChaCha and ChaCha with FAST Enhancement). Experimental results have discovered that the current projected has less encrypting time and better encrypting influence.

How to cite this article: A. R. Alawi and N. F. Hassan " A Proposal Video Encryption Using Light Stream Algorithm" Engineering and Technology Journal, Vol. 39, Part B, No. 01, pp. 184-196, 2021.

DOI: https://doi.org/10.30684/etj.v39i1B.1689

1. INTRODUCTION

The need for protecting the secret information from the un-authorized users has been led to applying a lot of encryption approaches. However, there is high importance in ciphering the multimedia contents which should be transmitted [1].

Encryption is applied to many types of media such as images, audio and video. Image encryption schemes have been progressively more studied to protect images from illegal access via providing secure image transmission over communication channels as reported in previous studies [2] and [3].

The need for performing the security requirements related to the digital speech signals resulted in developing regarding excellent encryption approaches as found in other work [4].
Regarding digital domain, the distribution networks are required to address two major issues:

1) Reducing huge communication requirements related to the multimedia data.
2) Protecting the copyrighted multimedia data.

Certain solution to the first issue was offered via the effective coding methods for video, audio, and images while the other issue was tackled in the privately defined closed systems through managing of the access to the Copyrighted Content (CC) [5].
Now, a main problem is protecting intellectual property regarding multimedia content in the multimedia networks.
In the digital video transmissions, the approaches of encryption were required for protecting digital videos from the attacks. Because the digital videos have huge sizes, they were typically transmitted in the compressed video formats like H.264/AVC, or MPEG. Therefore, encryption algorithms related to digital video should work in compressed domain. Many encryption algorithms of securing the video streaming were suggested where most of them have attempted on optimizing the process of encryption in terms of displaying the process and encryption speed [6].
Video encrypting algorithms are classified into five categories according to the approach of encryption information which are fully (completely) layered encryption, scrambling base encrypting, selective encryption, perceptual encryption, and chaotic encryption [7]. As explained from this classification, it has been proved that the completely layered video encryption has produced the highest level of video security, but it was computationally expensive because of its slow nature in processing the very large volume of video data and has in return limited its use in video encryption [8].

2. LITERATURE REVIEW

Enormous content of data transferred along the network have made video encryption very significant subject, but when video encryption works are reviewed, it is observed that only few works are focused on this subject. The following works present the most important video encryption algorithms:

Batham, et al. (2014) have presented Indexed Chaotic Sequence Based Selective Encryption of Compressed Video (ICSECV) by exploiting the compacted video's properties. Moreover, ICSECV is specifically scrambling the compacted Intra coded outlines as well as predictively coded outlines from each Gather of Pictures (GoP). Such encrypting approach has been found to be effective since it is providing real-time encryption related to the digital multimedia contents. Furthermore, to encrypt the selected frames that are not leaking information and providing the required security ICSECV utilized indexed-based chaotic sequence [9].

Hamood and Ibrahem (2018) have proposed video encryption method by using chaotic system for key generator and stream cipher where they have used chaotic map as one-time key generator which produced encryption key. Two approaches were proposed to generate key where the first approach uses cat map as generated key while the second approach presents larger key space because it uses three initial values where two for cat map and one for logistic map which has increased the number of initial values and the number of equations leading to increase the key generation time. Experimental results have shown that both proposed approaches are secure, and the reconstructed video is perfect with MSE equal to zero and maximum value for PSNR [10].

Malladar and Kunte (2019) have projected a work to emphasizing encrypting just the piece of the frame which was of greater attention of Video on Demand (VoD). Chosen Novel video encryption based on entropy was measured with the use of distinctive parameters such as correlation coefficient, PSNR, NPCR, and Histogram. The results have provided almost the optimum values and used for applications of the approach in VoD [11].

In the year 2020, a study conducted by Cheng, et al. have suggested an encryption approach for the H.264/AVC, H.264/AVC encrypting procedure encoding the video to many pieces with the use of Cipher Feedback (CFB) mode of AES with dynamic key. Furthermore, the key has been updated in real-time and generated through PRNG. The structure of encryption goes through 3 stages. Also, novel 4-D hyper chaotic algorithm has been suggested for protecting the data privacy. Experimental findings showed that the suggested technique for video encryption had less encryption time and better encryption effect [12].

3. FEATURES DETECTORS METHODS

The methods of feature extraction have high importance in image processing required for a lot of applications like object tracking, template matching, pattern recognition, and so on. The major aim of this approach is to reduce the image's information in efficient way and to maintain the significant information, and to increase the speed of the execution process where the approaches of features detectors might be divided into FAST Feature Detection Operator, Harris Corners Detector and Smallest Univalue Segment Assimilation Nucleus (SUSAN) [13]. Rosten and Drummondn cited in previous study [14] have suggested method of Features from Accelerated Segment Test (FAST) which is the corner detector and was applied to extract the image's corners. Such detector is based on the properties of feature standard related to AST. Also, the AST approach presents Bresenham ring of sixteen pixels round recent pixel. In the case when collection of n contiguous pixels that all have high darkness in comparison to current pixel minus threshold value (Ip − t), or at high brightness in comparison to such pixel (specified via Ip) with threshold value t (Ip + t). Afterwards, p was specified as corner and such approach has been defined as rapid feature extraction process [15].

4. LIGHT STREAM ALGORITHMS

One of the major categories related to symmetric encryption is stream cipher where it is encrypting byte or singular bit of the native data. Stream cipher's keys used for the process of encryption is randomly altered. Thus, the generated cipher was extremely hard to breach [16], [17]. Majorly, the stream ciphers are classified into two regions which are hardware and software backgrounds [18]. The ciphering and deciphering processes related to stream cipher is shown in the following equations:

$$C[s] = O[s] \oplus K[s] \quad (1)$$

$$O[s] = C[s] \oplus K[s] \quad (2)$$

where $\oplus$ represents mod by 2, C [s] represents cipher data bits, K [s] represents key random series bits, O [s] represents original data bits, while S represents 1 bit at the same time. With regard to the equations 1 and 2, the ciphering and deciphering together necessitate using identical seed key for producing identical keystream series K [S] [19]. It has been indicated that the stream cipher's security is depending majorly on key stream generator [20], [21] There are several light stream algorithms, such as Salsa and ChaCha but from our particle work, it's found that ChaCha (20) is suitable to be used in video encryption and the following is a description of ChaCha light stream cipher:

I. ChaCha Algorithm

This algorithm is developed by D.J. Bernstein (year) and defined as a modification related to the Salsa cipher. Also, it is ChaCha-8 one of the 256-bit stream ciphers based on 8-round cipher Salsa20/8. The modification from the Salsa20/8 algorithm to the ChaCha-8 improves the diffusion for each one of the rounds and conjecturally elevates resistance to the cryptanalysis whereas it keeps and sometimes enhances time per round. This algorithm follows comparable principles of design as the Salsa-20 algorithm where it provides excellent overall speed in comparison to the Salsa-20 for same security level [22] since ChaCha-8 consists of 8 rounds. Regarding the symmetric encryption, ChaCha has been used via Google and it is specified as one of the inheritor procedures with the alteration in the principal procedure of Salsa (20). With regards to the keystream generator, ChaCha uses identical calculation process. Yet, the first alteration was the matrix seeds which was changed as it can be seen in the figure 1 [23].

c1	c2	c3	c4		x0	x1	x2	x3
k1	k2	k3	k4	=	x4	x5	x6	x7
k5	k6	k7	k8		x8	x9	x10	x11
b1	b2	n1	n2		x12	x13	x14	x15

Figure 1: ChaCha Algorithm Input

The Second alteration is in Quarter Round Function (QRF) input. QRF of ChaCha is informing the input in the column's formula followed by diagonals formula as shown in Figure 2 [23].

Column form	Row form
QR(x 0, x 4, x 8, x 12)	QR(x0, x 5, x 10, x 15)
QR(x 1, x 5, x 9, x 13)	QR(x 1, x 6, x 11, x 12)
QR(x 2, x 6, x 10, x 14)	QR(x 2, x 7, x 8, x 13)
QR(x 3, x 7, x 11, x 15)	QR(x 3, x 4, x 9, x 14)

Figure 2: ChaCha Algorithm Four-Quarter Functions

The Third alteration is in QRF formula of ChaCha as shown in Figure 3 [23].

$$
\begin{aligned}
a &= a + b, \quad d = (d \oplus a) \lll 16 \\
c &= c + d, \quad b = (b \oplus c) \lll 12 \\
a &= a + b, \quad d = (d \oplus a) \lll 8 \\
c &= c + d, \quad b = (b \oplus c) \lll 7
\end{aligned}
$$

Figure 3: Procedures of ChaCha Algorithm in Four-Quarter Functions.

5. THE PROPOSED ALGORITHM

In this paper, a new cryptography was suggested to encrypt digital video file, where the video was encrypted by using ChaCha algorithm to provide speed and accuracy to encrypt video. The proposal encryption algorithms are consisted of following modules:

1) Entirely Encryption and Decryption Module
2) Partial Encryption and Decryption Module
3) Enhancement Partial Encryption and Decryption Module

I. Entirely Encryption and Decryption Module

In this section, encryption was done by applying encrypted digital video entirely. This phase is divided into phases:

A. Entirely Encryption Phase

In this phase, video to be encrypted was marked as input video and this input video was cut into several frames so as to process each frame separately, by entering this frame as an input to the ChaCha algorithm to be encrypted where encryption process consumed several seconds for each frame. This was involved by 8 rounds regarding the mathematical calculations with the use of XOR where the rotation and addition were utilized as inputs 4-byte constants, random key of 32byte, counter of 4byt, also 12byte nonce (Bernstein majorly defined the nonce and counter lengths for

being 8). Also, 4byte constants were 0x61707865, 0x3320646e, 0x79622d32, and 0x6b206574. In the ChaCha-20 algorithm, such strings have been concatenated. Counter, that is usually starting at 0 or 1 increments for each one of the 64-byte plaintext blocks. Figure 4 shows the block diagram of this phase, and Figure 5 illustrates the algorithm of entirely encryption digital video and algorithm of ChaCha-quarter function mentioned in previous study [23].

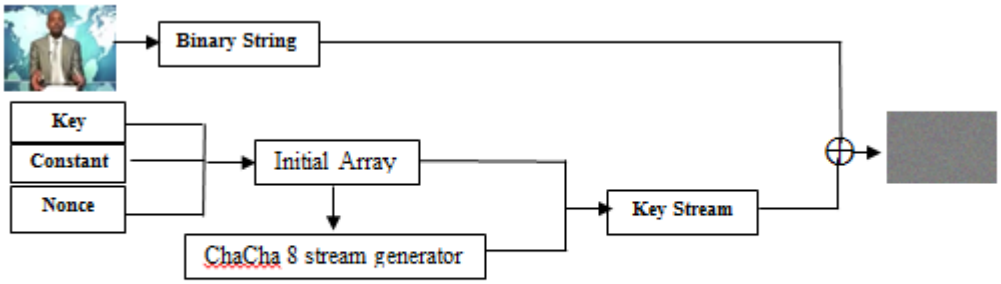

Figure 4: Block Diagram of Entirely Encryption Phase.

Algorithm(1): Entirely Encrypt Digital Video

Input: Digital Video File
Output: Encrypted Digital Video File

Begin
Step 1: Open video as a file.
Step 2: While not (EOF) do
Read video frame by frame. // Plain frame
Apply ChaCha encryption algorithm // ChaCha encryption algorithm.
Using Key K, Counter C, and Nonce N to create initial matrix X.
For i ← 0 to number of rounds -1 do
Apply the equations in figure (2)
End For
Merge all encrypted frames to create encrypted video.
End while
End

Figure 4: Entirely Encrypt Phase

B. Entirely Decryption Phase

In this phase, decryption was applied on encrypted video, i.e., the video produced from the encryption module is now decrypted in reverse process. At the beginning, encrypted video was divided into produced frames and each frame was decrypted entirely by using ChaCha with estimation decryption time shows in Table I, Table II and Table III. As all Encryption algorithms, in order to retrieve the original video, the encrypted frame was entered as an input to the decryption algorithm for restoring the original frame and then the process of combining the frames was done to restore the original video. All these operations have required relatively short time. Figure 6 shows block diagram of this phase while Figure 7 illustrates algorithm of entirely decryption digital video.

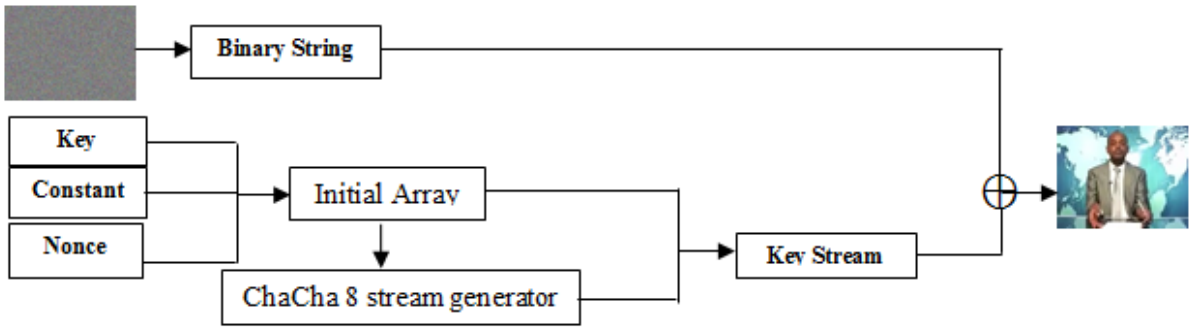

Figure 5: Block Diagram of Entirely Decryption Phase

Algorithm(2): Entirely Decrypt Digital Video

Input: Encrypted video file
Output: Original video file
Begin
Step1: Open video as a file.
Step2: While not (EOF) do
- Read video frame by frame.
- Apply ChaCha decryption algorithm.

End while
Step3: Merge all decrypted frames to create plain video.
End

Figure 6: Entirely Decrypt Digital Video Algorithm

II. Partial Encryption Digital Video Frames Module

Partial Encryption and Decryption Module was divided into phases:

A. Partial Encryption Phase:

In this phase, encryption was done partially which means encryption was applied on number of affected vision pixels in a frame. Feature operator to detect corners was used since these corners were candidate to be encrypted instead of encrypted whole frame for the purpose of reducing time. From several experiments, it was found that FAST operator was extremely adequate for the real-time applications of video processing due to its high-speed performance and thus it was adopted as a pre-processing before frames encrypted by ChaCha algorithm. This operator has produced a set of key points that caused a difference in the video. Regarding this detection approach, the candidate points have been specified via utilizing segment test for each of the frame pixels through specifying a circle of sixteen pixels around corner candidate pixel as computation's base. FAST was compared with the number existing detectors and it was found that FAST was not invariant to the scale changes with no robustness to the noise. Also, it was depending on threshold, in which choosing suitable threshold was not insignificant task. Figure 8 shows block diagram regarding partial encryption and Figure 9 displays Partial Encryption.

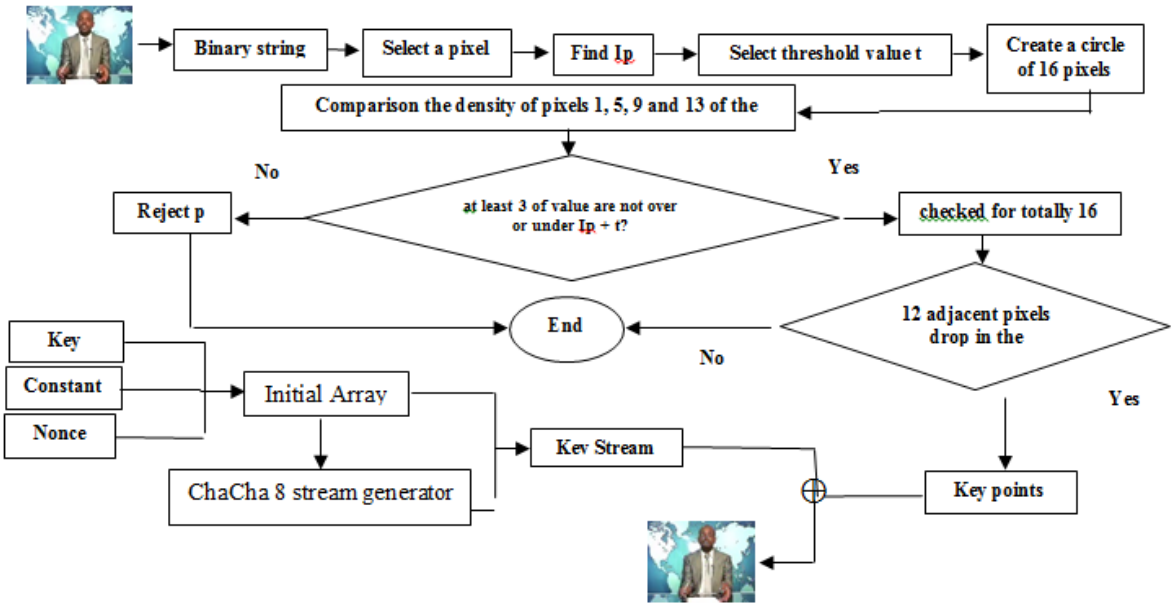

Figure 7: Block Diagram of Partial Encryption using ChaCha with FAST

Algorithm(3): Partial video encryption using ChaCha with FAST

Input: Video file
Output: Encrypted video
Begin
Step 1: Open video as a file.
Step 2: while not (EOF) do

- Read video frame by frame.
- Find key points using Features from Accelerated Segment Test.
- For each frame do:
  - Choice a pixel p in the image which is to be recognized as an importance point or not. Let its density be Ip.
  - Choice suitable threshold value t.
  - Consider a ring of 16 pixels round the pixel below test.
  - Here and now the pixel p is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than Ip + t, or all darker than Ip - t.
  - To make the algorithm fast, first compare the intensity of pixels 1, 5, 9 and 13 of the circle with Ip. As evident from the figure above, at minimum three of these four pixels should satisfy the threshold condition so that the importance point will exist.
  - If at minimum 3 of the four-pixel values — I1, I5, I9, I13 are not over or under Ip + t, then p is not an importance point (corner). In this situation discard the pixel p as a possible importance point. Else if at minimum 3 of the pixels are over or under Ip + t, then checked for totally 16 pixels and checked if 12 adjacent pixels drop in the condition.
  - Replication the process for totally the pixels in the frame
- End for.

End while
Step3: Apply ChaCha Encryption algorithm on the result from previous step.
Step4: Merge all encrypted frames to create encrypted video.
End

Figure 8: Partial video encryption algorithm using ChaCha with FAST

B. Partial Decryption Phase:

In this phase, decryption was done in reverse mode. Figure 10 shows block diagram of partial decryption while Figure 11 illustrates partial video decryption algorithm using ChaCha with FAST operator.

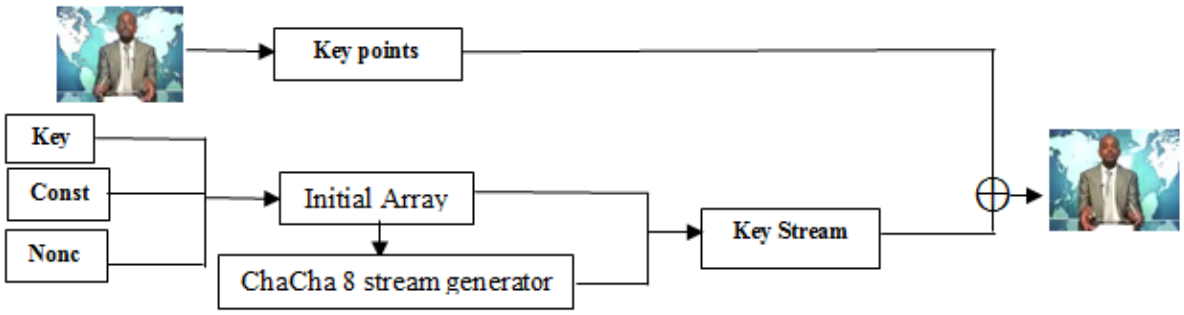

Figure 9: Decryption Flowchart using ChaCha with FAST

Algorithm(4): Partial video decryption using ChaCha with FAST

Input: Encrypted Video file
Output: Original video file
Begin
Step1: Open video as a file.
Step2: While not (EOF) do
- Read video frame by frame.
- Apply ChaCha decryption algorithm on key points.

End while
Step3: Merge all decrypted frames to create plain video.
End

Figure 10: Partial video decryption algorithm using ChaCha with FAST

III. Enhancement Partial Encryption and Decryption Module

Enhancement partial encryption and decryption module were divided into phases:

A. Enhancement Partial Encryption Phase:

To increase the invisibility of encrypted frame, extra set of adjacent points were added to be encrypted. Partial video encryption algorithm with added key points is illustrated in Algorithm 6 and number of whole points are shown in Table IV. Figure 12 shows block diagram of enhancement partial encryption and Figure 13 shows enhancement partial encryption algorithm.

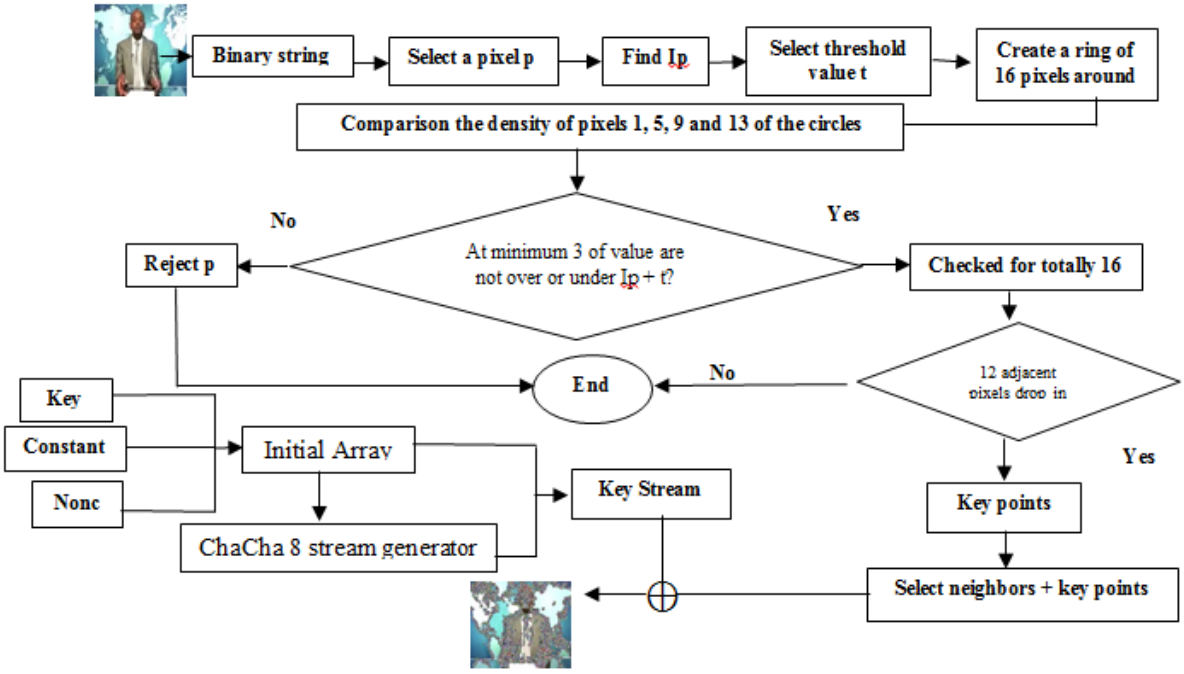

Figure 11: Block Diagram of Enhancement Partial Encryption

Algorithm (5): Enhancement Partial video Encryption

Input: Video file
Output: Encrypted video
Begin
Step 1: Open video as a file.
Step 2: while not (EOF) do

- Read video frame by frame.
- Find key points using Features from Accelerated Segment Test.
  - For each frame do:
  - Choice a pixel p in the image which is to be recognized as an importance point or not. Let its density be Ip.
  - Choice suitable threshold value t.
  - Consider a ring of 16 pixels round the pixel below test.
  - Here and now the pixel p is a corner if there exists a set of n adjacent pixels in the ring (of 16 pixels) which are all brighter than Ip + t, or all darker than Ip - t.
  - To make the algorithm fast, first comparison the density of pixels 1, 5, 9 and 13 of the ring with Ip. As obvious from the figure above, at minimum 3 of these four pixels should fulfill the threshold condition so that the importance point will exist.
  - If at minimum 3 of the four-pixel values — I1, I5, I9, I13 are not over or under Ip + t, then p is not an importance point (corner). In this case discard the pixel p as a possible importance point. Else if at minimum 3 of the pixels are over or under Ip + t, then checked for totally 16 pixels and checked if 12 adjacent pixels drop in the condition.
  - Construct matrix of 3*3 from contiguous pixel to importance point Ip (corner), in attempted to make encrypted frame more difficult.
  - Replication the process for totally the pixels in the frame.
  - End for.

End while
Step3: Apply ChaCha Encryption algorithm on the result from previous step.
Step4: Merge all encrypted frames to create encrypted video.
End

Figure 12: Enhancement Partial Video Encryption algorithm with added key points

B. Enhancement Partial Video Decryption Phase:

In this phase, the same operations as in encryption phase were done but in reverse order. Figure 14 shows block diagram of enhancement partial decryption while Figure 15 presents enhancement partial decryption algorithm.

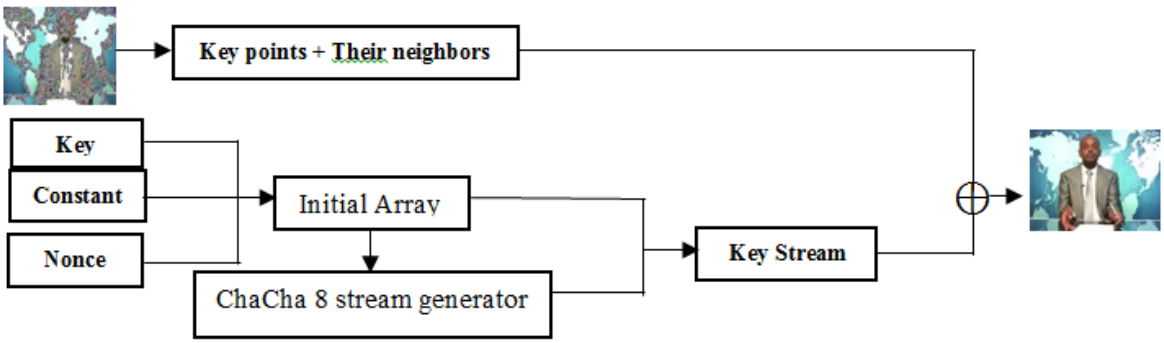

Figure 13: Block Diagram of Enhancement Partial Decryption

Algorithm(6): Enhancement Partial video Decryption

Input: Encrypted Video file
Output: Original video file
Begin
Step1: Open video as a file.
Step2: While not (EOF) do
- Read video frame by frame.
- Apply ChaCha decryption algorithm on key points.

End while
Step3: Merge all decrypted frames to create plain video.
End

Figure 14: Enhancement Partial video Decryption algorithm with added key points

6. EVALUATION EXPERIMENTAL RESULTS

Apparently, there was an importance of the key to the encryption process and its great role in increasing security and strengthening the algorithm and making it robust against the attackers. Below is Table I showing NIST key test and most of these tests (Bold) were good, as it is known, random and the enciphered image was impossible to be reconstructed if the key is differed by a small value.

TABLE I: Values of NIST Tests

Test #	Test name	P-Value of ChaCha encryption	# sub-tests
1.	Frequency Test	0.1691314447026715	Random
2.	Frequency Test within a Block	0.20636166380601706	Random
3.	Run Test	0.7107095740070324	Random
4.	Longest run of ones in a Block	0.418080498915231	Random
5.	Binary Matrix Rank Test	-1.0	Non-Random
6.	Discrete Fourier Transform (Spectral) Test	0.8185458083820408	Random
7.	Non-Overlapping Template Matching Test	0.9999833932213484	Random
8.	Overlapping Template Matching Test	nan	Non-Random
9.	Maurer's Universal Statistical test	-1.0	Non-Random
10.	Linear Complexity Test	-1.0	Non-Random
11.	Serial test:	0.4989610874592239 0.49853075529672125	Random Random
12.	Approximate Entropy Test	1.0	Random
13.	Cumulative Sums (Forward) Test	0.1398178398062382	Random
14.	Cumulative Sums (Reverse) Test	0.3381887424663557	Random

Speed in encryption operations was very important issue and should be considered when video is encrypted. In this research, 3 samples of the video were encrypted where the encryption was applied by using three suggested algorithms. Tables II, III, and IV show these three video samples sizes with entirely encryption and decryption time for five frame samples belong to each video.

TABLE II: Full Encryption time for five frames to each video samples

Video size	Frame Name	Encryption Time per seconds	Decryption Time per seconds
1.37 MB	Frame 1	11.923	11.676
	Frame 2	12.135	10.298
	Frame 3	11.383	10.253
	Frame 4	10.995	11.509
	Frame 5	12.122	10.656
3.16 MB	Frame 1	17.154	14.628
	Frame 2	14.598	14.310
	Frame 3	15.411	14.533
	Frame 4	15.121	14.039
	Frame 5	15.309	14.991

5.27 MB	Frame 1	22.337	18.871
	Frame 2	20.410	18.246
	Frame 3	22.120	19.508
	Frame 4	21.915	18.940
	Frame 5	21.024	17.899

TABLE III: Five frames with their encryption time using partial encryption

Video size	Frame Name	Number of Key Points	Encryption Time per seconds	Decryption Time per seconds
1.37 MB	Frame 1	6936	0.269	0.223
	Frame 2	6930	0.269	0.205
	Frame 3	6884	0.275	0.231
	Frame 4	7031	0.319	0.223
	Frame 5	7015	0.223	0.197
3.16 MB	Frame 1	1945	0.077	0.062
	Frame 2	1994	0.092	0.079
	Frame 3	2028	0.081	0.065
	Frame 4	2318	0.102	0.092
	Frame 5	2311	0.097	0.087
5.27 MB	Frame 1	3282	0.147	0.102
	Frame 2	3372	0.123	0.111
	Frame 3	3387	0.124	0.098
	Frame 4	3412	0.147	0.097
	Frame 5	3397	0.129	0.105

TABLE IV: Five frames with their encryption time using enhancement partial encryption

Video size	Frame Name	Number of Key Points	Encryption Time per seconds	Decryption Time per seconds
1.37 MB	Frame 1	6936	1.095	0.896
	Frame 2	6930	1.187	1.905
	Frame 3	6884	2.306	0.874
	Frame 4	7031	1.151	0.976
	Frame 5	7015	1.139	0.922
3.16 MB	Frame 1	1945	0.322	0.44
	Frame 2	1994	0.686	0.521
	Frame 3	2028	0.783	0.578
	Frame 4	2318	0.837	0.659
	Frame 5	2311	0.960	0.319
5.27 MB	Frame 1	3282	0.500	0.955
	Frame 2	3372	1.328	0.824
	Frame 3	3387	1.096	0.434
	Frame 4	3412	0.550	0.421
	Frame 5	3397	0.530	0.428

From above tables, it seems clear that the use of the ChaCha algorithm separately has given better results for each measure and the reason was due to the encrypting entire frame that gave better results since all pixels' values were changed, but this method has consumed time compared to the partial encryption method where only certain parts of the image were encrypted. So, the total difference in the pixel values was less, and this does not mean that it was less powerful than its predecessor, but certainly, it means much less needed time, and this the main purpose of the current proposal.

Based on visual inspection, it was not only adequate to judge the video encryption. Thus, other methods of measuring were needed for evaluating the degree of encryption quantitatively. Quality factors (MSE, PSNR, Correlation, NPCR, UACI and entropy) [24] were utilized for evaluating and comparison was made between the 2 suggested encryption algorithms which gave good result in encryption process.

The suggested encryption algorithms have given a good result not only in encryption time but also in encryption process. These values are illustrated in the Table V applied with three videos of different sizes:

TABLE V: Fidelity Criteria

Video size	Quality factors	Encryption using ChaCha	Encryption using Partial Encryption	Encryption using ChaCha with Enhancement FAST
1.37 MB	MSE	26328.318	1825.219	3924.521
	PSNR	3.926	15.517	12.192
	Correlation	-0.0018	0.90	0.79
	NPCR	99.60	64.04	68.43
	UACI	27.77	3.31	5.78
	Entropy	7.2571 Original 7.7065 Decrypt	7.2571 Original 7.3189 Decrypt	7.2571 Original 7.3626 Decrypt
3.16 MB	MSE	15744.076	149.917	355.018
	PSNR	6.159	26.372	22.628
	Correlation	-0.0013	0.96	0.92
	NPCR	99.47	24.19	28.45
	UACI	20.19	0.48	0.83
	Entropy	٧.٦٠٢٠ Original ٧.٧٠٦٦ Decrypt	7.6020 Original ٧.٦٠٠٣ Decrypt	7.6020 Original ٧.٥٩٩٣ Decrypt
5.27 MB	MSE	26983.749	190.041	410.426
	PSNR	3.819	25.342	21.998
	Correlation	-0.0004	0.98	0.96
	NPCR	99.55	14.79	16.87
	UACI	29.21	0.43	0.75
	Entropy	٦.٢٣٢٥ Original ٧.٧٠٦٣ Decrypt	6.2325 Original ٦.٢٥٠٠ Decrypt	٦.٢٣٢٥ Original ٦.٢٧٢١ Decrypt

From above table, the PSNR obtained values are low, and the MSE values are high, thus it shows that the proposed algorithm was good and robust against attacks. In the proposed algorithm, the correlation values were good. The values of NPCR and UACI showed that the algorithm was very resistive to differential attacks. The entropy was very close to a perfect value of 8, which shows that the proposed encryption algorithm has randomized the pixels well in the encrypted frame.

7. CONCLUSION

In this proposal, ChaCha encryption algorithm was used to encrypt digital video, since it is one of the lightest and fastest algorithms for encryption. ChaCha algorithm was used firstly, then hybrid algorithm that combined from the ChaCha with FAST algorithm to obtain fast encryption results. In addition, a proposed was an adjustment to the FAST to be adapted with our basic proposal by adding an extra set of adjacent points to be encrypted. Experimental results show a difference in time execution between the three proposed encryption algorithms, and the quality factors were considered to evaluate the proposed encryption algorithms. Results refer to good results in encryption time and quality of encryption process. In addition, there was an importance of the key to encryption process and its great role in increasing security and strengthening the algorithms leaving them robust against the attackers.

References

[1] M. F. Al-Jabali, "Image Encryption System by Generating Chains from the Secret Key." Middle East University, 2016.

[2] Y. H. Ali and H. A. Ressan, "Image Encryption Using Block Cipher Based Serpent Algorithm," Eng. Technol. J., vol. 34, No. 2 Part (B) Scientific, pp. 278-286, 2016.

[3] Y. H. Ail and Z. A. H. Alobaidy, "Images Encryption Using Chaos and Random Generation," Eng. Technol. J., vol. 34, No. 1 Part (B) Scientific, pp. 172-179, 2016.

[4] H. B. A. Wahab and S. I. Mahdi, "Speech Encryption Based on Wavelet Transformation and Chaotic Map," Eng. Technol. J., vol. 34, No. 5 Part (B) Scientific, pp. 721--729, 2016.

[5] N. R. M. S. Deeb, "Selective Encryption of Images Using Differential Evolution," Sel. Encryption Images Using Differ. Evol., 2011.

[6] M. Abomhara, O. Zakaria, and O. O. Khalifa, "An overview of video encryption techniques," Int. J. Comput. Theory Eng., Vol. 2, No. 1, p. 103, 2010.

[7] N. Geetha and D. K. Mahesh, "Efficient Video Encryption using RRS Algorithm," Int. J. Pure Appl. Math., Vol. 118, No. 9, pp. 885--890, 2018.

[8] D. Hooda and P. Singh, "A comprehensive survey of video encryption algorithms," Int. J. Comput. Appl., Vol. 59, No. 1, 2012.

[9] S. Batham, V. K. Yadav, and A. K. Mallik, "ICSECV: An efficient approach of video encryption," In 2014 Seventh International Conference on Contemporary Computing (IC3) (pp. 425-430).

[10] M. K. Ibrahem and L. A. Hamood, "Video Encryption Based on Chaotic System and Stream Cipher," Iraqi Journal of Information & Communications Technology, Vol. 1, Issue (2), 33-40. (2018).

[11] R. Malladar and R. S. Kunte, "Selective Video Encryption Based on Entropy Measure," in Integrated Intelligent Computing, Communication and Security, Springer, 2019, pp. 603--612.

[12] S. Cheng, L. Wang, N. Ao, and Q. Han, "A Selective Video Encryption Scheme Based on Coding Characteristics," Symmetry (Basel)., Vol. 12, No. 3, p. 332, 2020.

[13] G. Kumar and P. K. Bhatia, "A detailed review of feature extraction in image processing systems," in 2014 Fourth international conference on advanced computing & communication technologies, 2014, pp. 5--12.

[14] M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly, "Image features detection, description and matching," in Image Feature Detectors and Descriptors, Springer, 2016, pp. 11--45.

[15] Y. Wu, "Research on feature point extraction and matching machine learning method based on light field imaging," Neural Comput. Appl., Vol. 31, No. 12, pp. 8157-8169, 2019.

[16] M. Bakhtiari and M. A. Maarof, "An efficient stream cipher algorithm for data encryption," Int. J. Comput. Sci. Issues, vol. 8, No. 3, p. 247, 2011.

[17] M. Stamp, "Information security: principles and practice," New York: Wiley, vol. 2, 2011.

[18] A. Jolfaei and A. Mirghadri, "Survey: image encryption using Salsa20," Int. J. Comput. Sci. Issues, vol. 7, No. 5, p. 213, 2010.

[19] Y. Minglin and M. Junshuang, "Stream ciphers on wireless sensor networks," in 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 2011, vol. 3, pp. 358--361.

[20] P. P. Deepthi, D. S. John, and P. S. Sathidevi, "Design and analysis of a highly secure stream cipher based on linear feedback shift register," Comput. Electr. Eng., Vol. 35, No. 2, pp. 235--243, 2009.

[21] T. Good and M. Benaissa, "Hardware results for selected stream cipher candidates," State art stream ciphers, vol. 7, pp. 191--204, 2007.

[22] P. Yadav, I. Gupta, and S. K. Murthy, "Study and analysis of eSTREAM cipher Salsa and ChaCha," in 2016 IEEE International Conference on Engineering and Technology (ICETECH), 2016, pp. 90--94.

[23] D. J. Bernstein, "ChaCha, a variant of Salsa20," in Workshop Record of SASC, 2008, vol. 8, pp. 3-5.

P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius and T. Blažauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic - Tent map," Entropy, 21 (7), 656, 2019.