BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 1
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0004

A New Symmetric Digital Video Encryption Model

Krasimir Kordov, Georgi Dimitrov

Konstantin Preslavski University of Shumen, 9712 Shumen, Bulgaria.
E-mails: krasimir.kordov@shu.bg g.dimitrov@shu.bg

Abstract*: *In this paper a new symmetric cryptographic method for digital video file is presented. The proposed algorithm is based on combining two chaotic maps. Extended cryptographic is provided for evaluation and proving the efficiency and the level of security of the encrypted files. The empirical tests are explained and the obtained results are presented in the manuscript.

Keywords*: *Video encryption, Symmetric encryption, Cryptography, Cryptographic analysis, Digital video files.

1. Introduction

Cryptography is a tool that helps to establish information security used to modify data in unreadable form that can only be restored with the correct secret key. This is a secure method for secret communication, protecting the information. Cryptographic analysis has the opposite purpose and its goal is to analyze the encryption in order to restore the plain information.

In modern days cryptographic algorithms have evolved significantly because the information is mostly digitally stored in computer systems in the form of binary sequences. During the past few decades researchers have been exploring the possibilities of securing digital files for safe keeping and transferring them in network communications.

Some of the most used files in computer systems, social networks and streaming platforms are the digital video files. That is the reason why video encryption algorithms have to be developed for copyright protection and payments for video streaming.

The goal of this paper is to design an encryption algorithm for digital video files. In order to achieve this goal, the structure of that specific file type is analyzed and we use the standard frame processing approach, assuming the video files being composed by sequences of static images. The object of our study is raw video format because there is no compression and data loss during both the encryption and decryption processes, which allows us to perform an extensive cryptographic analysis in the form of empirical experiments and comparisons.

The related works in this field of research give us a starting point for our study. Quao and Nahrstedt [1] present video encryption algorithm with dynamic secret key usage in frame processing and Deshmukh and Kolhe [2] present improved variant of AES (Advanced Encryption Standard) encryption demonstrating there is a way of improving the known cryptographic algorithms. Yang and Sun [3] have designed video encryption based on chaotic logistic maps, demonstrating better results with that method. Another example of digital video security in real time processing is presented in [4]. The cryptographic analysis in these papers demonstrates quality encryption and the results are used for comparison. Since frame processing in video files is similar to image processing, we also compare our results with [5, 6] and some of the latest research presented in [7, 8, 9] and shortly described in Table 1.

Table 1. Description of encryption algorithms

Reference	Method of encryption	Comments
[1]	Encryption with dynamic secret key for frame processing	Designed for compressed video encryption. Advantages: strong encryption
[2]	Improved AES encryption	Designed for video encryption. Advantages: improved performance in encryption
[3]	Frame scrambling and encryption with chaotic maps	Designed for real time video encryption. Advantages: strong encryption
[4]	Improved Hill Encryption Algorithm	Designed for real time video encryption. Advantages: strong encryption
[5]	Rubik's cube method for encryption	Designed for image encryption Advantages: fast encryption
[6]	Pixel Shuffling and BASE 64 Encoding with Logistic map	Designed for image encryption. Advantages: increased key-space
[7]	Walsh-Hadamard transform and Arnold and Tent maps	Designed for image encryption. Advantages: increased key-space
[8]	Modified zigzag transformation and key generation using enhanced logistic Tent Map	Designed for image encryption. Advantages: strong encryption
[9]	Fractional-order edge detection and generalized chaotic maps	Designed for image encryption. Advantages: strong encryption

The method proposed in this paper is realized with the MATLAB software. The test video files are selected with different characteristics (such as size, length, frames, frames per second, etc.) for more reliable results. The empirical tests are used for extensive cryptographic analysis including visual comparing, histogram comparing, adjacent pixels comparing, etc. All values are obtained using MATLAB except for some of the statistical tests that are performed using specific software, described later in this paper.

2. Pseudo-random generator combining Hitzl-Zele map and Tinkerbell map

The Pseudo-Random Number Generators (PRNG or PRG) are cryptographic primitives used for stream encryption. They are software realized at low cost and their purpose is to produce random binary sequences. Examples can be found in [10].

The PRGs are often based on chaotic maps, because of their chaotic behavior and high sensitivity to the initial conditions [11, 12]. These advantages benefit the cryptographic systems, because the PRGs provide endless binary sequences needed for both the encryption and decryption processes and the sensitivity of the initial conditions, which is used for the secret keys' evaluation.

2.1. Three-dimensional Hitzl-Zele map

Hitzl and Zele [13] have explored the properties of the two-dimensional quadratic map invented by Hénon and proposed their three-dimensional version. Saha and Strogatz [14] have made an additional research concerning the chaotic behavior of the three-dimensional variant. The Hitzl-Zele map is analytically determined by the next equation:

$$\begin{aligned} x_{i+1} &= 1 + y_{\mathrm{i}} - z_{\mathrm{i}} x_{\mathrm{i}}^2, \\ y_{i+1} &= a x_i, \\ z_{i+1} &= b x_i^2 + z_i - 0.5, \end{aligned} \tag{1}$$

where the bifurcation parameters *a* and *b* are set to 0.25 and 0.87, respectively, for chaotic behavior. The graphical representation is shown in Fig. 1.

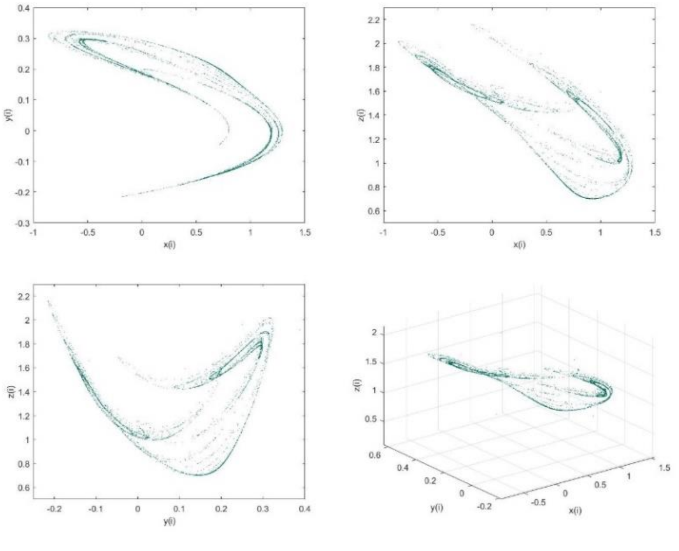

Fig. 1. Plot of the Hitzl-Zele map with all combinations of *x*, *y* and *z* dimensions

2.2. Tinkerbell map

The Tinkerbell map [15-17] is another chaotic map often used in cryptography and it is given by

$$\begin{aligned} x_{t+1} &= x_t^2 - y_t^2 + a x_t + b y_t, \\ y_{t+1} &= 2 x_t y_t + c x_t + d y_t, \end{aligned} \tag{2}$$

where the parameters have specific fixed values for chaotic behavior: *a*= 0.9, *b* = –0.6013, *c* = 2.0, and *d* = 0.50. The graphical representation is shown in Fig. 2.

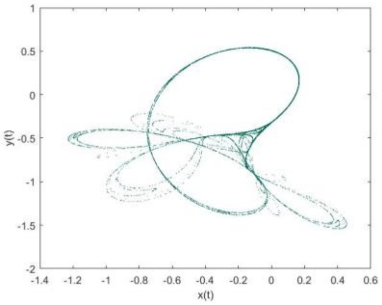

Fig. 2. Plot of the Tinkerbell map using *x* and *y* dimensions

2.3. Random bits extraction scheme

The proposed PRG performs the following steps:

- The initial values from (1) and (2) are determined. We have used *x*=911346643, *y*=0.632467713, *z*=0.097523107 for (1) and *x*=–0.145622309, *y*=–0.742799703 for (2), based on our previous studies. The parameters are as described in the equations presentation.
- The Hitzl-Zele map is iterated *N* times and the Tinkerbell map is iterated *M* times.
- From the next iteration of the Hitz-Zele map, the temporary value temp1 is obtained – temp1 = abs((int)(z_i .10^7)) mod2.
- From the next iteration of the Tinkerbell map, the temporary variable temp2 is obtained – temp2 = abs((int)(x_t .10^8)) mod2.
- The current random bit is extracted by performing XOR operation between the variables temp1 and temp2.
- The previous three steps are repeated until the needed binary sequence is reached.

2.4. Randomness evaluation

The Pseudo-random generators provide binary sequences, but to determine whether produced bits are random a further statistical analysis is required. The most used statistical software packages are DIEHARD Test Software [18] and NIST Statistical Test Suite [19]. Both test packages require an input sequence of at least 1 billion bits for reliable results. DIEHARD software performs 19 tests for randomness evaluation of the produced binary sequence and for every test to be successfully passed the obtained *P*-value needs to be in the range [0, 1). The second software NIST performs 17 tests for randomness evaluation. Again the obtained *P*-value needs to be in the range [0, 1) and in addition the input binary sequence is divided to 1000 subsequences of length of 1 million bits each. The minimum pass rate for the 15 statistical tests is approximately 980 (from 1000 binary sequences) and for the last two tests (Random excursions and Random excursions variant) the minimum pass rate for the random excursion (variant) test, is approximately 595 (from 609 binary sequences). The results are presented in Table 2 and Table 3 and clearly show that every test is passed which is an indication that the proposed PRG is secure enough to be used in cryptographic algorithms.

Table 2. DIEHARD statistical tests results

Test	P-value	Result
Birthday spacing	0.5258407	Pass
Overlapping 5-permutation	0.2817740	Pass
Binary rank (31×31)	0.4888080	Pass
Binary rank (32×32)	0.6899190	Pass
Binary rank (6×8)	0.5395834	Pass
Bitstream	0.4175650	Pass
OPSO	0.5661826	Pass
OQSO	0.4783500	Pass
DNA	0.4702806	Pass
Stream count-the-ones	0.6718850	Pass
Byte count-the-ones	0.5277914	Pass
Parking lot	0.4917039	Pass
Minimum distance	0.2729040	Pass
3D spheres	0.6693350	Pass
Squeeze	0.3405600	Pass
Overlapping sums	0.6500550	Pass
Runs up	0.5463990	Pass
Runs down	0.5154980	Pass
Craps	0.4771900	Pass

Table 3. NIST statistical tests results

Test	P-value	Pass rate	Result
Frequency (monobit)	0.655854	985/1000	Pass
Block-frequency	0.962688	990/1000	Pass
Cumulative sums (Forward)	0.699313	987/1000	Pass
Cumulative sums (Reverse)	0.856359	987/1000	Pass
Runs	0.486588	996/1000	Pass
Longest run of Ones	0.219006	989/1000	Pass
Rank	0.353733	991/1000	Pass
FFT	0.678686	991/1000	Pass
Non-overlapping templates	0.476852	990/1000	Pass
Overlapping templates	0.397688	988/1000	Pass
Universal	0.498313	993/1000	Pass
Approximate entropy	0.729870	987/1000	Pass
Random-excursions	0.594951	604/609	Pass
Random-excursions Variant	0.511341	604/609	Pass
Serial 1	0.749884	994/1000	Pass
Serial 2	0.270265	991/1000	Pass
Linear complexity	0.657933	996/1000	Pass

2.5. Key-space and key-sensitivity analysis

The important requirement for the PRGs is a security concern – the secret key must have key-space larger than 2^{100} to resist brute-force attacks. The key-space is defined by the initial variables because their values combinations and variations include all possible secret keys. For the proposed PRG, the initial double variables from (1) are $x_{i(0)}$, $y_{i(0)}$ and $z_{i(0)}$ and from (2) – $x_{t(0)}$ and $y_{t(0)}$. Considering the IEEE floating point double variables standard total key-space for the proposed PRG is $10^{15\times 5} \approx 2^{249}$ plus $2^{32\times 2} = 2^{64}$ for the integer variables N and M. The total key-space is approximately 2^{313}, which is secure enough.

The other important requirement for PRGs is the key sensitivity. To evaluate the behavior of the proposed PRG, an experiment is performed with very similar but different secret keys (the initial values of the variables). The secret Key 1 (K1) uses the values from Subsection 2.3; for K2 $x_{i(0)}$ is changed to 0.911346644; for K3 $y_{i(0)}$ = 0.632467714; for K4 $z_{i(0)}$ = 0.097523108; for K5 $x_{t(0)}$ = –0.145622308; for K6 $y_{t(0)}$ is changed to –0.742799704. The result sequences from this experiment are shown in Fig. 3.

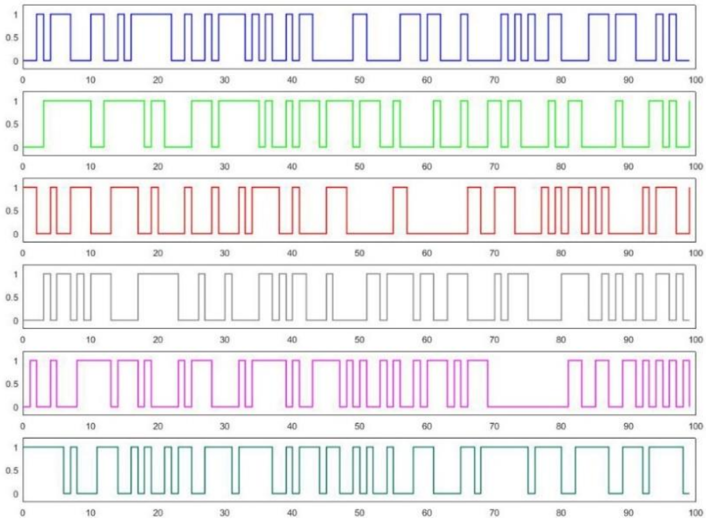

Fig. 3. Key-sensitivity analysis

The comparison in binary sequences shown in Fig. 3 demonstrates that the proposed PRG is highly sensitive to any changes in the initial conditions.

3. Digital video encryption model

In using the PRG (described in Section 2) as a cryptographic primitive for digital video encryption the main step is to process the video file as a composition of frames that need to be secured. Frame processing is performed by treating the frame as digital image, where every pixel has color value that is modified with bit-stream produced by the PRG and using XOR operation. The encryption scheme is demonstrated in Fig. 4.

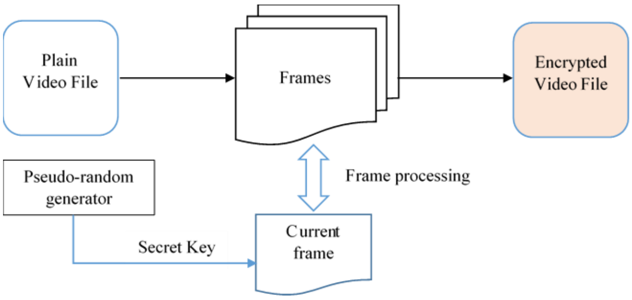

Fig. 4. Digital video encryption scheme

The proposed cryptographic is symmetric meaning the decryption process requires the same steps and mandatory using of the exact same secret key. Visual example (first frames) is shown in Fig. 5.

Fig. 5. Example video file with its corresponding encrypted and decrypted file

4. Cryptographic analysis

Proving the security of cryptographic algorithms requires extensive cryptographic analysis. For the empirical experiments five videos have been tested (encrypted and decrypted) with the proposed scheme. Random frames from the test videos have been selected for further evaluation and comparison and the results are presented in this section.

4.1. Visual and histogram analysis

The main purpose of visual analysis is to determine if there are any traces (objects and colors) of the original file after the encryption process. All the experiments are similar to those of Fig. 5 and show that encrypted files don't have any similarity with the plain files which is an indication of strong cryptography.

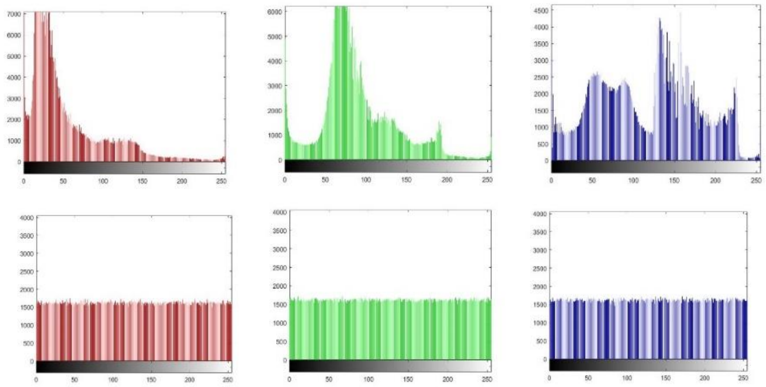

Fig. 6. Histogram analysis – red (left), green (middle) and blue (right) channels

Histogram analysis is another method for comparison between the plain file and the corresponding encrypted file and shows the tonal distribution of the colors in frames. Fig. 6 represents histograms of color distribution for red, green and blue color channels of a frame form the plain file and corresponding video file. Comparing the results histograms clearly show that the encrypted files have unified color distribution and have no similarity with the plain files.

4.2. Information entropy

As part of the cryptographic analysis, the entropy is used for measurement of the uncertainty in the information theory. In this case we applied this test to determine the probability of certain pixel value appearance in the video frames. The information entropy is calculated with the formula

$$H(X) = -\sum_{i=0}^{N} p(x_i)\log_2 p(x_i), \tag{3}$$

where: X is a variable; $p(x_i)$ is a function of the probability of x to have a certain value of x_i; N is 255 because the colors values (i) of every pixel of the frame is from 0 to 255 for every color of the RGB scheme. The best value for information entropy is $H(X) = 8$, for truly chaotic systems. The experiment is performed by testing five of the frames from our test digital files. In Table 4 are presented the obtained results compared with other algorithms.

Table 4. Information entropy analysis

Plain file	Frame No	Entropy	Encrypted file	Frame No	Entropy
Video1.avi	1	7.6464004	Video1.avi	1	7.9998447
Video1.avi	240	7.5924187	Video1.avi	240	7.9998582
Video2.avi	1	7.5799054	Video2.avi	1	7.9996889
Video2.avi	153	7.5993386	Video2.avi	153	7.9996241
Video3.avi	1	6.6717239	Video3.avi	1	7.9997355
Video3.avi	380	6.5025463	Video3.avi	380	7.9996928
Video4.avi	1	7.7214311	Video4.avi	1	7.9998720
Video4.avi	244	7.7187708	Video4.avi	244	7.9998560
Video5.avi	1	6.8938548	Video5.avi	1	7.9998324
Video5.avi	72	6.9277413	Video5.avi	72	7.9998470
Ref. [4]	–	6.234655	–	–	7.997266
Ref. [6]	–	7.4318	–	–	7.9968
Ref. [8]	–	7.2730	–	–	7.9993
Ref. [9]	–	7.4455	–	–	7.9993
Ref. [2]	–	–	–	–	7.941
Ref. [7]	–	–	–	–	7.9980

The results in Table 3 show that the entropy of the encrypted file is very close to the perfect value 8, meaning the color values of the pixels are chaotic, because of the strong encryption process.

4.3. Correlation coefficient analysis

This test is designed to evaluate the similarity between the adjacent pixels in the frames. Normally the plain images have similar colors of their adjacent pixels, which means the neighbor pixels' colors can be restored. The good encryption algorithms leave no similarities between the colors of the adjacent pixels in the frames. The correlation coefficient is calculated by the equation

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{4}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2,$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2,$$

$$\operatorname{cov}(x,y) = \sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}) .$$

In (4) x_i and y_i are the adjacent pixel color values, $\bar{x}$ and $\bar{y}$ are the mean values, N is the number of the pixel couples, cov(x, y) is the covariance for calculation of the correlation coefficient r_{xy}.

The correlation values are always in the range [–1, 1]. Values close to |±1| mean strong correlation (the color are similar and dependent) and values close to 0 mean weak correlation (the values are completely different and there is no dependency between the values). The results from our experiment are presented in Table 5.

Table 5. Correlation coefficient analysis

Plain file / Frame No	Direction	Correlation coefficient	Encrypted file/ Frame No	Direction	Correlation coefficient
Video1.avi Frame 1	Horizontal Vertical Diagonal	0.965734111 0.952241534 0.925129767	Video1E.avi Frame 1	Horizontal Vertical Diagonal	–0.000310817 –0.000762672 0.000322516
Video1.avi Frame 240	Horizontal Vertical Diagonal	0.977757210 0.966742446 0.951297432	Video1E.avi Frame 240	Horizontal Vertical Diagonal	–0.000018004 0.000214075 0.001491880
Video2.avi Frame 1	Horizontal Vertical Diagonal	0.991226282 0.984790500 0.981214278	Video2E.avi Frame 1	Horizontal Vertical Diagonal	–0.001647818 –0.001655260 –0.000539033
Video2.avi Frame 153	Horizontal Vertical Diagonal	0.994451733 0.988626670 0.986545861	Video2E.avi Frame 153	Horizontal Vertical Diagonal	–0.000271109 –0.000799092 –0.000310437
Video3.avi Frame 1	Horizontal Vertical Diagonal	0.979000256 0.946560125 0.937573864	Video3E.avi Frame 1	Horizontal Vertical Diagonal	–0.000134484 –0.000359892 0.001197012
Video3.avi Frame 380	Horizontal Vertical Diagonal	0.990826997 0.973203333 0.967805990	Video3E.avi Frame 380	Horizontal Vertical Diagonal	0.000728419 –0.000987650 0.002088115
Video4.avi Frame 1	Horizontal Vertical Diagonal	0.980078909 0.992098908 0.974391618	Video4E.avi Frame 1	Horizontal Vertical Diagonal	–0.000185231 –0.000265785 –0.000570940
Video4.avi Frame 244	Horizontal Vertical Diagonal	0.982051602 0.992247133 0.976870276	Video4E.avi Frame 244	Horizontal Vertical Diagonal	0.000540990 –0.000818698 –0.001293385
Video5.avi Frame 1	Horizontal Vertical Diagonal	0.979450816 0.988765397 0.973027405	Video5E.avi Frame 1	Horizontal Vertical Diagonal	–0.000490464 0.000031310 0.000498608
Video5.avi Frame 72	Horizontal Vertical Diagonal	0.980318039 0.991817070 0.974511808	Video5E.avi Frame 72	Horizontal Vertical Diagonal	–0.000146780 –0.000126860 0.001895751
Ref. [2]	Horizontal Vertical Diagonal	0.9452 0.9471 0.9127	–	Horizontal Vertical Diagonal	–0.0112 –0.0813 0.0009
Ref. [3]	Horizontal Vertical Diagonal	0.9671 0.9655 0.9683	–	Horizontal Vertical Diagonal	0.00251 0.00237 0.00198
Ref. [8]	Horizontal Vertical Diagonal	0.9505 0.9745 0.9668	–	Horizontal Vertical Diagonal	–0.0237 –0.0178 –0.0284
Ref. [9]	Horizontal Vertical Diagonal	0.9719 0.9850 0.9639	–	Horizontal Vertical Diagonal	0.0028 0.00097633 0.00003127

Results in Table 5 represent the correlation coefficients of the first and the last frames of the tested digital video files. All encrypted files have values very close to 0, which means the adjacent pixels' values have no dependence, indicating strong encryption.

4.4. Number of pixels change rate

The Number of Pixels Change Rate (NPCR) is an indicator that measures the difference between plain and encrypted files. This test compares the corresponding pixel values of the same frame from a plain file against an encrypted file and shows the percentage difference between the two files. NPCR is calculated as follows:

(5)
$$\mathrm{NPCR} = \frac{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1} D(i,j)}{W \times H} \times 100\%,$$
$$D(i,j) = 1 \text{ (if } x_{i,j} \neq y_{i,j}\text{)}, D(i,j) = 0 \text{ (if } x_{i,j} = y_{i,j}\text{)}.$$

In (5) $x_{i,j}$ and $y_{i,j}$ are the corresponding pixel from both files. The results of this test are presented in Table 6.

Table 6. NPCR analysis

Plain File	Encrypted File	Frame No	NPCR, %
Video1.avi	Video1E.avi	1	99.605918852880663
Video1.avi	Video1E.avi	240	99.602864583333329
Video2.avi	Video2E.avi	1	99.608258928571431
Video2.avi	Video2E.avi	153	99.613095238095241
Video3.avi	Video3E.avi	1	99.603587962962962
Video3.avi	Video3E.avi	380	99.616174768518519
Video4.avi	Video4E.avi	1	99.605918852880663
Video4.avi	Video4E.avi	244	99.598926183127574
Video5.avi	Video5E.avi	1	99.605918852880663
Video5.avi	Video5E.avi	72	99.614519032921805
Ref. [5]	–	–	99.5850
Ref. [6]	–	–	99.6149
Ref. [8]	–	–	99.6017

The results in Table 6 demonstrates that the difference between the plain and encrypted video files is always greater than 99.5% which is an indicator that the encryption alters entirely the result files.

4.5. Computational and complexity analysis

The proposed algorithm is tested with MATLAB software with 2.40 GHz Intel ® Core™ i7-3630QM Dell Inspiron laptop (middle class computer system). The complexity of the proposed algorithm is defined by the computations and iterations of the encryption/decryption calculations. Considering the linear computation of every iteration, for pixel encryption and decryption of every frame, the total complexity of every frame is $\Theta(n^2)$ meaning the proposed algorithm depends on the rows and columns of every frame (frame width and frame height) and also it depends on the number of frames.

The selected test video files are with different size, length, number of frames, frames per second, etc. and the results are presented in Table 7.

Table 7. Encryption/decryption time

Video No	Video 1	Video 2	Video 3	Video 4	Video 5
Frames	250	167	400	246	97
Frames per 1 second	25 fps	30 fps	29 fps	50 fps	50 fps
Frame width	720	560	640	720	720
Frame height	576	320	360	576	576
Video size (KB)	303.823 KB	87.745 KB	270.007 KB	298.963 KB	117.923 KB
Video length (s)	00:00:10	00:00:05	00:00:13	00:00:04	00:00:01
Encryption/ Decryption time (per 1 frame)	456 s	197 s	251 s	453 s	452 s

5. Conclusion

The manuscript describes a model for digital video files encryption, which is based on secure pseudorandom generator. The PRG is designed using two chaotic maps and the further analysis indicates enough security levels to be the basis of a cryptographic system. The key-space analysis shows good resistance against brute-force attacks, key sensitivity analysis shows high sensitivity for the initial conditions (the secret key) and the randomness evaluation demonstrates the produced binary sequences are random.

The digital video encryption scheme is using the proposed PRG and frame by frame processing for the final encryption. The cryptographic analysis evaluates the encryption method by empirical tests. The visual and the histogram analysis show no visual traces comparing the plain and encrypted video files. The NPCR test confirms that we have more than 99.5% difference in analyzed corresponding files and the correlation coefficient analysis demonstrates that adjacent pixels have always different color values in the encrypted files, unlike the plain ones. The information entropy analysis shows chaotic distribution in color values of the result encrypted files confirming the strong encryption process.

Acknowledgments**:** This work is supported by the European Regional Development Fund and the Operational Program “Science and Education for Smart Growth” under contract UNITe No BG05M2OP001-1.001-0004-C01 (2018-2023).

References

1. Qiao, L., K. Nahrstedt. A New Algorithm for MPEG Video Encryption. – In: Proc. of 1st International Conference on Imaging Science System and Technology, 1997, pp. 21-29.
2. Deshmukh, P., V. Kolhe. Modified AES Based Algorithm for MPEG Video Encryption. – In: International Conference on Information Communication and Embedded Systems (ICICES’14), 2014, pp. 1-5.
3. Yang, S., S. Sun. A Video Encryption Method Based on Chaotic Maps in DCT Domain. – Progress in Natural Science, Vol. **18**, 2008, No 10, pp. 1299-1304.

4. Yang, T., Y. Li, C. Lai, J. Dong, M. Xia. The Improved Hill Encryption Algorithm Towards the Unmanned Surface Vessel Video Monitoring System Based on Internet of Things Technology. – Wireless Communications and Mobile Computing, 2018.
5. Loukhaoukha, K., J. Y. Chouinard, A. Berdai. A Secure Image Encryption Algorithm Based on Rubik's Cube Principle. – Journal of Electrical and Computer Engineering, Vol. **7,** 2012.
6. Sathishkumar, G. A., K. B. Bagan. A Novel Image Encryption Algorithm Using Pixel Shuffling and Base 64 Encoding Based Chaotic Block Cipher (IMPSBEC). – WSEAS Transactions on Computers, Vol. **10**, 2011, No 6, pp. 169-178.
7. Sneha, P. S., S. Sankar, A. S. Kumar. A Chaotic Colour Image Encryption Scheme Combining Walsh-Hadamard Transform and Arnold-Tent Maps. – Journal of Ambient Intelligence and Humanized Computing, Vol. **11**, 2020, No 3, pp. 1289-1308.
8. Ramasamy, P., V. Ranganathan, S. Kadry, R. Damaševičius, T. Blažauskas. An Image Encryption Scheme Based on Block Scrambling, Modified Zigzag Transformation and Key Generation Using Enhanced Logistic-Tent Map. – Entropy, Vol. **21** 2019, No 7, 656.
9. Ismail, S. M., L. A. Said, A. G. Radwan, A. H. Madian, M. F. Abu-ElYazeed. A Novel Image Encryption System Merging Fractional-Order Edge Detection and Generalized Chaotic Maps. – Signal Processing, Vol. **167**, 107280.
10. Kordov, K. M. Modified Chebyshev Map Based Pseudo-Random Bit Generator. – In: AIP Conference Proceedings, Vol. **1629**, 2014, pp. 432-436.
11. Kordov, K. M. Modified Pseudo-Random Bit Generation Scheme Based on Two Circle Maps and XOR Function. – Applied Mathematical Sciences, Vol. **9**, 2015, No 3, pp. 129-135.
12. Kordov, K. M. Signature Attractor Based Pseudorandom Generation Algorithm. – Advanced Studies in Theoretical Physics, Vol. **9**, 2015, No 6, pp. 287-293.
13. Hitzl, D. L., F. Zele. An Exploration of the Hénon Quadratic Map. – Physica D: Nonlinear Phenomena, Vol. **14**, 1985, No 3, pp. 305-326.
14. Saha, P., S. H. Strogatz. The Birth of Period Three. – Mathematics Magazine, Vol. **68**, 1995, No 1, pp. 42-47.
15. Goldsztejn, A., W. Hayes, P. Collins. Tinkerbell is Chaotic. – SIAM Journal on Applied Dynamical Systems, Vol. **10**, 2011, No 4, pp. 1480-1501.
16. Alefeld, G., J. Herzberger. Introduction to Interval Computations, Comput. – In: Sci. Appl. Math. New York, Academic Press, 1983.
17. Alligood, K. T., T. D. Sauer, J. A. Yorke. Chaos: An Introduction to Dynamical Systems. Berlin, Springer-Verlag, 1996.
18. Marsaglia, G. The Marsaglia Random Number CDROM Including the Diehard Battery of Tests of Randomness. Tallahassee, FL, USA, Florida State University, 1995.
19. Rukhin, A., et al. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application. NIST Special Publication 800-22. Gaithersburg, MD, USA, NIST, 2001.

Received: 19.07.2020; Second Version: 17.11.2020; Accepted: 21.12.2020