Preprint. By permission. Equivalent in content, but not in format, to the final publication in *Modelling and Simulation 2025 – Proc. European Simulation and Modelling Conference, ESM'2025*, published by EUROSIS – The European Multidisciplinary Society for Modelling and Simulation Technology.
Full citation:
L. J. Chmielewski, A. Orłowski, M. Nieniewski. Both shares in color visual cryptography can be statistically indistinguishable from noise. In *Modelling and Simulation 2025 – Proc. European Simulation and Modelling Conference, ESM'2025*, pp. 77–81. Ghent, Belgium, 22-24 October, 2025. EUROSIS – The European Multidisciplinary Society for Modelling and Simulation Technology. https://www.eurosis.org/cms/index.php?q=taxonomy/term/58#ESM.
Last compiled on October 23, 2025.

Both Shares in Color Visual Cryptography Can Be Statistically Indistinguishable from Noise

Leszek J. Chmielewski[1]
[1]Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland
leszek_chmielewski@sggw.edu.pl

Arkadiusz Orłowski[1,2]
[2]Institute of Mathematics and Cryptology, Military University of Technology – WAT, Warsaw, Poland
arkadiusz.orlowski@wat.edu.pl

Mariusz Nieniewski[3]
[3]Faculty of Mathematics and Informatics, University of Lodz, Łódź, Poland
mariusz.nieniewski@wmii.uni.lodz.pl

Keywords
Visual cryptography, Color, Randomness, NIST tests.

Abstract

In visual cryptography, each share must appear statistically indistinguishable from noise. The first share is generated using a pseudorandom number generator; the second one is constructed based on the first one and the image to be encoded. This is true for black-and-white, gray, and color versions of the scheme. In previous work, we proposed a method for generating the second share that preserves essential encoding constraints while aiming to maintain the statistical randomness of the first share. To assess the randomness of shares by applying typical randomness tests (NIST STS) we simulate hundreds of share samples and analyze the resulting p-value distributions. This simulation enables us also to test the uniformity of distributions visually. In this study, we extend our assessment by testing for deviation of these distributions from uniformity using Kolmogorov–Smirnov and chi-squared tests. Results of simulations indicate that our method preserves randomness in the second share to a degree comparable to that of the first one, supporting its use in statistically robust visual encryption of color images.

Introduction

Randomness is a cornerstone of cryptographic security. It underpins not only key generation and protocol unpredictability, but also structural concealment in schemes such as visual cryptography (VC). In VC, a secret image is divided into multiple binary *shares* which reveal the secret image when overlayed and viewed by bare eye. Shares, when viewed individually, must reveal no meaningful information. This concealment depends not merely on logical irreversibility, but on statistical indistinguishability. In fully random cryptography schemes each share must resemble uniform random noise. Other possibilities are that shares simply represent no useful information, or mimic images which contain irrelevant views (Dhiman and Kasana 2018).

In classical schemes, the first share is expected to be random. A physical random process could be used to this aim; however, in practice the random process is simulated with a pseudorandom number generator (PRNG). The second share is computed to two separately desirable targets. First it is of a utmost importance to satisfy the reconstruction constraints to a largest possible extent. Second, the simulation of randomness of the second share is also a target. As a result, while the first share reflects the statistical properties of the PRNG, the second one may contain artifacts introduced by the encoding algorithm. If these artifacts introduce patterns detectable by statistical means, the concealment may be compromised – not visually, but analytically. In previous work, we proposed a method for generating the second share that minimizes such distortion. Our construction modifies the second share in a controlled manner, introducing the information necessary for decoding, while attempting to maintain the original randomness. Informally, the second share should "look just as random" as the first one, even though it is not independently sampled.

Preliminary visual and histogram-based analyses (Chmielewski, Nieniewski, and Orłowski 2022a) suggested that this goal might be achievable: p-values from standard randomness tests (e.g., rank, longest run) applied to sets of shares were broadly well-distributed, with few values falling below conventional significance thresholds. However, such observations, while suggestive, lack formal statistical confirmation (cf. (James 1995)).

In this paper, we address this gap. For a given image, we simulate the process of multiple uses of the algorithm by generating 100 random instances of the first share (with PRNG) and derive corresponding second shares using our method. Each share, treated as a one-dimensional series of bits that represent pixels read by rows or by columns, is tested using a battery of randomness tests (NIST Statistical Test Suite, NIST STS (Bassham, Rukhin, Soto, Nechvatal, Smid, Leigh, Levenson, Vangel, Heckert, and Banks 2010)), and the resulting p-values are aggregated. We analyze these p-value vectors in two ways: (1) by counting how many

fall below a fixed rejection threshold ($\alpha = 0.01$) (as in (Chmielewski, Nieniewski, and Orłowski 2022a)) and (2) by testing their histogram for deviations from uniformity using the Kolmogorov–Smirnov and chi-squared tests. This second-order statistical analysis, i.e., treating p-value distributions as objects of inference, enables a more rigorous evaluation of the method's ability to preserve randomness. Our findings suggest that the second shares exhibit no statistically significant degradation in randomness relative to the first ones, indicating that a controlled transformation with random as well as deterministic elements need not compromise statistical concealment in visual cryptography.

Surprisingly enough, statistically testing the shares in color visual cryptography was not widely applied. In (Ulutas, Yazici, Nabiyev, and Ulutas 2008) shares were tested with the correlation-based tools. Recently, several authors (Toktas, Erkan, and Yetgin 2024; Tang, Lu, Zhang, Huang, Huang, and Wang 2024; Liu and Ding 2024) tested the security of key-based ciphers for images with NIST STS, but their methods had no relation to purely visual cryptography in which bare eye would be enough to reveal the secret image by overlaying the shares.

Visual coding algorithms

In first visual cryptography of black-and-white images each pixel of an image was represented by a 2×2 *tile*. The shares were not random but had a structure which made it possible to represent pixels at two levels: bright (half-white) and dark (black) (Naor and Shamir 1995). For representing images with more shades of grey the dithering techniques were used (see for example (Stinson and Paterson 2018)). For color images, also the dithering schemes with additive as well as subtractive basic colors were used (Yang and Chen 2008; Dhiman and Kasana 2018). We go along the same path, but as the starting point we take an entirely random tile, as far as the random number generator can be considered as actually simulating the randomness correctly. So, the tile is in no way crafted for the coding process.

The R, G, B and K pixels in it are taken form a random integer number generator in $\{0, 1, 2, 3\}$. Throughout this paper 6×6 tiles are used, each representing one color pixel in the coded secret, which is a compromise between the restored image quality and pixel expansion. For each pixel, hence a tile, the information on the numbers of R, G, B and K (blacK) pixels that need to be represented come from the dithering process: these are the reconstruction constraints. Now, there are two concepts of coding to meet these constraints. We shall concentrate on a single pixel in a pair of tiles, in share 1 and 2. The process is strictly pixelwise parallel.

Coding by hiding: the tiles in the two shares are originally identical, so all the pixels are initially uncovered. By swapping the pixels, by pairs, in the share 2, with the pairs selected random, covering the unnecessary pixels is attempted, to meet the reconstruction constraints as closely as possible (see (Chmielewski, Nieniewski, and Orłowski 2021) for details).

Coding by unhiding: the tile 2 is formed by choosing for each pixel a color different from that in tile 1, at random, so all the pixels are initially covered (other simple ways to get a random share 2 with all pixels covered exist). By swapping the pixels, by pairs, in the share 2, with the pairs selected at random, uncovering the necessary pixels is attempted to meet the reconstruction constraints as closely as possible (see (Chmielewski, Nieniewski, and Orłowski 2022b) for details).

Let us note that each color, not only black, can hide any other color, due to that overlaid color shares are transparent narrow-band filters.

The pixels in share 2 are only swapped, or randomly generated and swapped, so it is aimed to preserve their statistical randomness while encoding the secret image. In both algorithms errors occur due to that the tiles are random, not engineered for the errorless operation of the algorithms. There are not enough color pixels or some pixels cannot be covered due to lack of enough pixels in another color in share 2. An example of coding and decoding is shown in Figure 1 (fragment of a figure from (Chmielewski, Nieniewski, and Orłowski 2022a), according to licence). In coding by unhiding, the surplus bright pixels can be easily avoided, but the lack of enough color pixels to appear unhidden can occur more frequently than in the coding by unhiding (Chmielewski, Nieniewski, and Orłowski 2022a). For typical data, the coding by hiding is more appropriate, so this method will be considered further in this paper. In Figure 2 an example of decoding a natural image is shown. It can be marginally noticed that the quality attainable with bare eye, for example in field conditions, is very limited, which is typical for this class of decoding methods, while it is possible to greatly enhance the image quality with very simple computations (see (Chmielewski, Nieniewski, and Orłowski 2022a)).

Materials and methods

For each tested image, we generated 100 independent instances of the share 1 using a default pseudorandom number generator. Each corresponding share 2 was computed using the method described above. Both shares were treated as binary data and subjected to statistical randomness evaluation using a standard test suite (NIST STS). For each share, and for each test, we obtained a single p-value indicating the degree to which the observed result is consistent with the null hypothesis of randomness. This produced, per image and per

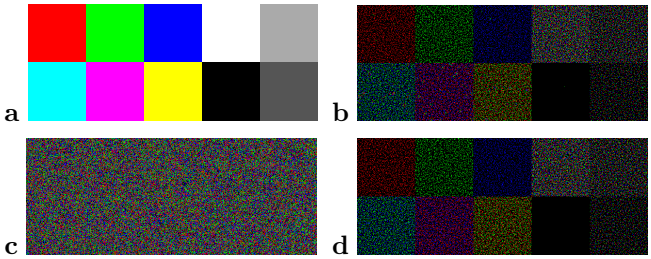

Figure 1: Decoding for a Test Image `test100c`. (**a**) Original, 100×40. (**b**) Decoded 600×240 with the *hiding* method. (**d**) Decoded 600×240 with the *unhiding* method. (**c**) Share 2, a random image

test, a vector of 100 p-values for Share 1 and 100 for Share 2, each in two versions: read by rows and read by columns, so there are four sets of data for each tested image. We tested six images frequently used as benchmarks: `baboon` (quarter edge size), `parrots` (half edge size), `peppers`, `Lena`, `Boats` and `test100c`, a small test image used in our previous papers (Chmielewski, Nieniewski, and Orłowski 2022a). There are 15 tests in the NIST STS, some with subtests, which gives 188 tests (default values of parameters were used, unless stated otherwise). In total, there were 4512 independent realizations, or samples, of p-value vectors, 100 elements each vector.
Two levels of analysis were performed on these samples of p-value vectors, each vector constituting a histogram:

Rejection count: the number of p-values falling below a fixed significance threshold ($\alpha = 0.01$). This test (p-v) quantifies how often a given test would flag a sample as non-random under standard criteria.

Uniformity tests: the distribution of p-values was tested for consistency with the uniform distribution $U(0, 1)$ using both the Kolmogorov–Smirnov (K-S) test and Pearson's chi-squared (χ^2) test. This meta-analysis treats the p-values themselves as data to evaluate second-level randomness.

Results

Across all images and tests, both share 1 and share 2 exhibited low rejection rates, with the number of p-values below 0.01 typically within the expected range under the null hypothesis. In particular, share 2 did not display systematically higher rejection counts than share 1.
Histogram-based uniformity assessments yielded similarly positive outcomes. In the majority of cases, neither the KS test nor the chi-squared test rejected the null hypothesis of uniformity of the p-value histogram at the 0.05 level. Where rejections did occur, they were scattered and statistically compatible with type I error rates. Figure 3 (fragment of a figure from (Chmielewski, Nieniewski, and Orłowski 2022a), according to licence) presents representative p-value histograms for a single test (`Rank`) across 100 share 1 and share 2 instances, each by rows and by columns, illustrating their visual similarity. More histograms, also for a method without the randomness property, can be seen in (Chmielewski, Nieniewski, and Orłowski 2022a).
Tables 1, 2 and 3 summarize p-v, K-S and χ^2 tests for some of the many possible image-test-share-direction combinations, confirming the absence of systematic rejections and their generally marginal counts among the 4512 samples of the p-value histograms.
Not all the tests revealed failures. Number of non-randomness cases detected by NIST tests are shown in Table 4 for both methods (test names were abbreviated to reduce table width). The `OverlappingTemplate` opens the ranking for both methods. Other tests revealed much less failures. It can be observed that some tests detected no failures at all; these were: `ApproximateEntropy`, `CumulativeSums 2`, `LinearComplexity`, `Rank`, `Runs`, `Universal`, and some subtests of `NonOverlappingTemplate`, `RandomExcursions` and `RandomExcursionsVariant`.
These results indicate that the mixed random and deterministic transformations applied to generate share 2 do not introduce detectable structure that would be revealed by typical randomness tests. In particular, the p-value distributions of share 2 are statistically indistinguishable from those of share 1, both in rejection frequency and in second-order uniformity.

Discussion

The primary aim of this study was to assess whether the transformation used to encode information in the second share of a visual cryptographic pair necessarily degrades the statistical appearance of randomness.
Our findings suggest that it does not. Across a wide range of images and tests, the second shares generated using our method passed standard randomness assessments at rates comparable to purely random first shares. Neither rejection counts nor p-value distributions indicated systematic anomalies.
These results provide evidence that a carefully designed mixed random and deterministic transformation can preserve the statistical noise characteristics essential for

Table 1: Rejections by Test

Test	Share 1	Share 2	Rows	Cols	Fails/Samps
Hiding method					
p-v	43	36	45	34	79/4512
K-S	16	29	16	29	46/4512
χ^2	19	27	23	23	46/4512
Unhiding method					
p-v	49	51	49	51	100/4512
K-S	16	20	21	15	36/4512
χ^2	24	25	28	21	49/4512

Figure 2: Decoding for a Natural Image `boats`. (**a**) Original, 787×576. (**b**) Decoded 4722×3456 with the *hiding* method. (**c**) Share 2, upper left quarter, a random image

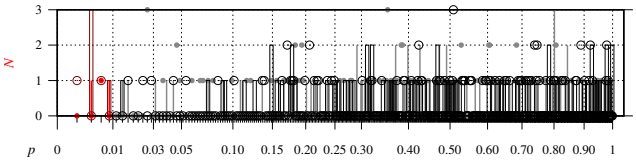

Figure 3: Histograms of 400 p-values for Test `Rank`, Image `boats`, *Hiding* Method. Marks: bright full: share 1, dark empty: share 2, circles: by rows, bars: by columns; red: 9 rejections total

Table 2: Rejections in Test Combinations

Test	Counts/4512 Samples		
Hiding method			
Single	p-v: 79	KS: 41	χ^2: 46
Pairs	p-v, K-S: 8	p-v, χ^2: 9	K-S, χ^2: 14
3-tuple	p-v, K-S, χ^2: 7		
Unhiding method			
Single	p-v: 100	KS: 36	χ^2: 49
Pairs	p-v, K-S: 9	p-v, χ^2: 9	K-S, χ^2: 13
3-tuple	p-v, K-S, χ^2: 7		

Table 3: Rejections by Shares/Directions

Case	Counts/4512 Samples		
Hiding method			
By shares	Share 1: 69	Share 2: 73	Both: 8
By directions	Rows: 74	Columns: 68	Both: 6
Unhiding method			
By shares	Share 1: 77	Share 2: 84	Both: 7
By directions	Rows: 83	Columns: 78	Both: 9

the security of visual cryptography. Importantly, this holds not only at the level of individual tests, but also in aggregate behavior: the p-value distributions themselves remain uniform, suggesting that the transformation does not bias or constrain the randomness landscape.

From a methodological standpoint, this study demonstrates the utility of meta-level randomness evaluation, i.e., treating p-value vectors as analyzable objects. This approach adds a layer of robustness beyond pass/fail summaries, enabling the detection of subtle statistical distortions that may not manifest as outright rejections. Finally, the use of two independent statistical environments (Python and R) for validation strengthens the reliability of the results and minimizes the risk of software-related bias.

Conclusion

In this paper, we analyzed whether the generation of the second share in visual cryptography compromises its statistical randomness. Using standard randomness tests and second-order evaluations of p-value distributions, generated in a simulated experiment, we found no evidence of such degradation. Our method for constructing the second share produces outputs that, despite being algorithmically constrained, remain statistically indistinguishable from noise. This confirms its suitability for secure visual encryption and demonstrates that controlled determinism can coexist with apparent randomness. More broadly, our work shows that meta-analysis of test outputs provides a powerful tool for validating the integrity of cryptographic structures under transformation.

References

Bassham, L. E., A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. Levenson, M. Vangel, N. A. Heckert, and D. L. Banks (2010, 16 Sep). A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, National Institute of Standards and Tech-

Table 4: Numbers of Failures (N) Revealed by NIST Tests

Hiding method			*Unhiding* method		
#	N	Test name, subtest	#	N	Test name, subtest
1	12	OverlapTempl	1	14	OverlapTempl
2	3	FFT	2	4	NonOvrlTempl 22
3	3	NonOvrlTempl 1	3	4	NonOvrlTempl 26
4	3	NonOvrlTempl 26	4	4	RandExcurs 8
5	3	NonOvrlTempl 30	5	3	NonOvrlTempl 67
6	3	NonOvrlTempl 73	6	3	NonOvrlTempl 73
7	3	NonOvrlTempl 75	7	3	NonOvrlTempl 79
8	3	NonOvrlTempl 135	8	3	NonOvrlTempl 81
9	2	BlockFreq	9	3	NonOvrlTempl 86
10	2	NonOvrlTempl 2	10	3	NonOvrlTempl 111
11	2	NonOvrlTempl 3	11	2	Frequency
12	2	NonOvrlTempl 12	12	2	FFT
13	2	NonOvrlTempl 13	13	2	NonOvrlTempl 25
14	2	NonOvrlTempl 15	14	2	NonOvrlTempl 29
15	2	NonOvrlTempl 25	15	2	NonOvrlTempl 31
16	2	NonOvrlTempl 32	16	2	NonOvrlTempl 33
17	2	NonOvrlTempl 40	17	2	NonOvrlTempl 34
18	2	NonOvrlTempl 45	18	2	NonOvrlTempl 35
19	2	NonOvrlTempl 48	19	2	NonOvrlTempl 42
20	2	NonOvrlTempl 71	20	2	NonOvrlTempl 66
21	2	NonOvrlTempl 76	21	2	NonOvrlTempl 84
22	2	NonOvrlTempl 81	22	2	NonOvrlTempl 85
23	2	NonOvrlTempl 90	23	2	NonOvrlTempl 87
24	2	NonOvrlTempl 95	24	2	NonOvrlTempl 95
25	2	NonOvrlTempl 109	25	2	NonOvrlTempl 97
26	2	NonOvrlTempl 110	26	2	NonOvrlTempl 102
27	2	NonOvrlTempl 119	27	2	NonOvrlTempl 103
28	2	NonOvrlTempl 131	28	2	NonOvrlTempl 106
29	2	NonOvrlTempl 136	29	2	NonOvrlTempl 117
30	2	NonOvrlTempl 137	30	2	NonOvrlTempl 135
31	2	RandExcurs 5	31	2	NonOvrlTempl 147
32	2	RandExcurs 8	32	2	RandExcVar 2
33	2	RandExcVar 5	33	2	RandExcVar 4
34	2	RandExcVar 6	34	2	RandExcVar 18
35	1	CumulatSums 1	35	2	Serial 2
36	1	LongestRun	36	1	NonOvrlTempl 1
37	1	Rank	37	1	NonOvrlTempl 3
38	1	NonOvrlTempl 5	38	1	NonOvrlTempl 6
39	1	NonOvrlTempl 7	39	1	NonOvrlTempl 11
40	1	NonOvrlTempl 9	40	1	NonOvrlTempl 13
41	1	NonOvrlTempl 10	41	1	NonOvrlTempl 14
42	1	NonOvrlTempl 11	42	1	NonOvrlTempl 18
43	1	NonOvrlTempl 18	43	1	NonOvrlTempl 19
44	1	NonOvrlTempl 20	44	1	NonOvrlTempl 21
45	1	NonOvrlTempl 31	45	1	NonOvrlTempl 24
46	1	NonOvrlTempl 36	46	1	NonOvrlTempl 27
47	1	NonOvrlTempl 37	47	1	NonOvrlTempl 28
48	1	NonOvrlTempl 38	48	1	NonOvrlTempl 40
49	1	NonOvrlTempl 39	49	1	NonOvrlTempl 41
50	1	NonOvrlTempl 43	50	1	NonOvrlTempl 48
51	1	NonOvrlTempl 51	51	1	NonOvrlTempl 49
52	1	NonOvrlTempl 52	52	1	NonOvrlTempl 50
53	1	NonOvrlTempl 56	53	1	NonOvrlTempl 51
54	1	NonOvrlTempl 58	54	1	NonOvrlTempl 52
55	1	NonOvrlTempl 60	55	1	NonOvrlTempl 55
56	1	NonOvrlTempl 65	56	1	NonOvrlTempl 56
57	1	NonOvrlTempl 66	57	1	NonOvrlTempl 57
58	1	NonOvrlTempl 77	58	1	NonOvrlTempl 58
59	1	NonOvrlTempl 82	59	1	NonOvrlTempl 59
60	1	NonOvrlTempl 83	60	1	NonOvrlTempl 60
61	1	NonOvrlTempl 84	61	1	NonOvrlTempl 61
62	1	NonOvrlTempl 85	62	1	NonOvrlTempl 62
63	1	NonOvrlTempl 87	63	1	NonOvrlTempl 74
64	1	NonOvrlTempl 88	64	1	NonOvrlTempl 75
65	1	NonOvrlTempl 91	65	1	NonOvrlTempl 76
66	1	NonOvrlTempl 92	66	1	NonOvrlTempl 77
67	1	NonOvrlTempl 93	67	1	NonOvrlTempl 80
68	1	NonOvrlTempl 94	68	1	NonOvrlTempl 82
69	1	NonOvrlTempl 96	69	1	NonOvrlTempl 83
70	1	NonOvrlTempl 97	70	1	NonOvrlTempl 93
71	1	NonOvrlTempl 98	71	1	NonOvrlTempl 96
72	1	NonOvrlTempl 102	72	1	NonOvrlTempl 99
73	1	NonOvrlTempl 107	73	1	NonOvrlTempl 101
74	1	NonOvrlTempl 111	74	1	NonOvrlTempl 104
75	1	NonOvrlTempl 114	75	1	NonOvrlTempl 107
76	1	NonOvrlTempl 115	76	1	NonOvrlTempl 109
77	1	NonOvrlTempl 116	77	1	NonOvrlTempl 113
78	1	NonOvrlTempl 117	78	1	NonOvrlTempl 114
79	1	NonOvrlTempl 120	79	1	NonOvrlTempl 116
80	1	NonOvrlTempl 123	80	1	NonOvrlTempl 118
81	1	NonOvrlTempl 127	81	1	NonOvrlTempl 120
82	1	NonOvrlTempl 134	82	1	NonOvrlTempl 122
83	1	NonOvrlTempl 138	83	1	NonOvrlTempl 126
84	1	NonOvrlTempl 142	84	1	NonOvrlTempl 127
85	1	NonOvrlTempl 146	85	1	NonOvrlTempl 128
86	1	RandExcurs 1	86	1	NonOvrlTempl 130
87	1	RandExcVar 1	87	1	NonOvrlTempl 131
88	1	RandExcVar 3	88	1	NonOvrlTempl 132
89	1	RandExcVar 4	89	1	NonOvrlTempl 134
90	1	RandExcVar 10	90	1	NonOvrlTempl 138
91	1	RandExcVar 11	91	1	NonOvrlTempl 141
92	1	RandExcVar 13	92	1	NonOvrlTempl 143
93	1	RandExcVar 14	93	1	NonOvrlTempl 145
94	1	RandExcVar 15	94	1	NonOvrlTempl 148
			95	1	RandExcurs 7
			96	1	RandExcVar 3
			97	1	RandExcVar 5
			98	1	RandExcVar 8
			99	1	RandExcVar 10
			100	1	RandExcVar 12
			101	1	RandExcVar 15
			102	1	Serial 1

nology, Gaithersburg, MD, USA. Series: Special Publication (NIST SP), Rep. No. 800-22 Rev 1a.

Chmielewski, L. J., M. Nieniewski, and A. Orłowski (2021, 28-30 Jun). Can color visual cryptography be truly random? In M. Choraś, R. S. Choraś, M. Kurzyński, et al. (Eds.), *Progress in Image Processing, Pattern Recognition and Communication Systems – Proc. Int. Conf. CORES, IP&C, ACS 2021*, Volume 255 of *LNNS*, Bydgoszcz, Poland, pp. 72–86. Springer, 2022.

Chmielewski, L. J., M. Nieniewski, and A. Orłowski (2022a, 19-21 Sep). Error analysis and graphical evidence of randomness in two methods of color visual cryptography. In L. J. Chmielewski and A. Orłowski (Eds.), *Computer Vision and Graphics: Proc. ICCVG 2022*, Volume 598 of *LNNS*, Warsaw, Poland, pp. 237–267. Springer, Cham, 2023.

Chmielewski, L. J., M. Nieniewski, and A. Orłowski (2022b, 25-27 Apr). Truly random color visual cryptography without surplus color spikes. In P. Jędrzejowicz and I. Czarnowski (Eds.), *Proc. 3rd Polish Conference on Artificial Intelligence PP-RAI'2022*, Gdynia, Poland, pp. 53–56. Publishing House of Gdynia Maritime University.

Dhiman, K. and S. S. Kasana (2018). Extended visual cryptography techniques for true color images. *Computers & Electrical Engineering 70*, 647–658.

James, F. (1995). Chaos and randomness. *Chaos, Solitons & Fractals 6*, 221–226.

Liu, S. and Q. Ding (2024). A new color image encryption algorithm based on the memristor hyperchaos system and Rubik's cube theory. *The European Physical Journal Plus 139*(9), 820.

Naor, M. and A. Shamir (1995, 9-12 May). Visual cryptography. In A. De Santis (Ed.), *Advances in Cryptology — EUROCRYPT'94. Proc. Workshop on the Theory and Application of Cryptographic Techniques*, Volume 950 of *LNCS*, Perugia, Italy, pp. 1–12. Springer.

Stinson, D. R. and M. Paterson (2018). *Cryptography: Theory and Practice* (4th ed.). Boca Raton: Chapman and Hall/CRC Press.

Tang, J., M. Lu, Z. Zhang, X. Huang, T. Huang, and J. Wang (2024). Novel asymmetrical color image encryption using 2D sine-power coupling map. *Nonlinear Dynamics 112*(13), 11547–11569.

Toktas, F., U. Erkan, and Z. Yetgin (2024). Cross-channel color image encryption through 2D hyperchaotic hybrid map of optimization test functions. *Expert Systems with Applications 249*, 123583.

Ulutas, M., R. Yazici, V. V. Nabiyev, and G. Ulutas (2008, 27-29 Oct). (2,2)-secret sharing scheme with improved share randomness. In *Proc. 23rd Int. Symp. Computer and Information Sciences ISCIS 2008*, Istanbul, Turkey, pp. 1–5. IEEE.

Yang, C.-N. and T.-S. Chen (2008). Colored visual cryptography scheme based on additive color mixing. *Pattern Recognition 41*(10), 3114–3129.